IQinVision Network API Manual

Release Date: August 9, 2011

Table of Contents

Notices

The release date for this manual is August 9, 2011. Due to updates in the camera's operating software, this manual will change periodically and may change without notice. Contact IQinVison for the most current version of this manual.

The screenshots in this manual are used for explanatory purposes. Your actual screens may differ from the screenshots included in this manual.

1. About This Manual

1.1. General

This IQinVision API manual contains the main functionality and API information required for host based connections to IQeye cameras. It also describes how to customize IQeye camera web pages, startup options and how to create a simple private labeled camera look and feel.

In addition this manual also includes documentation on how to use the IQeye Video System Management and Control ("vSMAC") advanced programmatic interface. This allows a very feature-rich host based interface with IQeye cameras. For example, the *Cameo* tab which is available on all IQinVision Pro Line cameras (IQeye7xx/8xx/Axx) demonstrates the vSMAC functionality of VCAMS whereby the user is able to create multiple video streams each with independent frame rate, resolution and area of interest.

Note: *Not all of the commands used by IQinVision cameras are documented in this manual. For a full description of all commands available from the IQeye command line please see the IQeye Reference Manual.*

2. Discovery and Control

2.1. Discovering IQeye cameras

Currently two discovery mechanisms are used to search for IQeye cameras from a network host. Pro Line cameras use IQinVision's proprietary GSAUCE protocol. Basic Line cameras use Zero Configuration Networking (Zeroconf).

Future firmware releases for the Pro Line will also support Zeroconf so that there will be a consistent mechanism for finding all IQinVision cameras. In addition, IQinVision will be providing a Windows DLL and sample source code that performs both GSAUCE and Zeroconf queries and will return to the caller a list of cameras found.

2.1.1. Zeroconf discovery protocol

Basic Line cameras use Zeroconf (www.zeroconf.org) as their discovery mechanism. There are several different mechanisms that host applications can use to find cameras using Zeroconf.

Host application programs can build the Zeroconf protocol right into an application based on the IETF draft documentation and sample code available from www.zeroconf.org.

Applications can also use a Zeroconf client which can provide an API to discover network devices. Sample clients include Bonjour (http://www.apple.com/macosx/technology/bonjour.html) and Avahi, http://avahi.org.

Basic line cameras will also self-assign a link-local IP address as defined in RFC 3927 if no other IP address has been configured. This allows host machines to discover these cameras with no additional configuration.

2.1.2. GSAUCE discovery protocol

Pro Line cameras use IQinVision's GSAUCE discovery protocol. GSAUCE is a UDP based protocol that will use broadcast packets to query for cameras on a local area network. Once cameras have been found authenticated UDP packets can be used to assign IP parameters. Please contact Engineering Support, EngineeringSupport@iqeye.com for the GSAUCE protocol definition and sample code.

2.2. IQdevice DLL

The IQdevice DLL supports camera discovery and basic camera manipulation functions such as assigning IP address, rebooting a camera, and setting and getting OIDs. Please contact Engineering Support, EngineeringSupport@iqeye.com to get this software.

3. Object Identifiers

All configurable options on IQeye cameras can be controlled using Object Identifiers. ("OIDs")

3.1. OID description

OIDs are SNMP-like Object IDentifiers that allow host based applications and on-camera web pages to get and set values on the camera. An object is a single piece of information that can be displayed and possibly changed, i.e. a flag setting, a string, a counter variable, etc.

An OID is represented by a dot separated numerical value. For example, the OID 2.4 corresponds to the name of the camera. OIDs are used in all built-in web pages and can also be used by host application programs to configure the camera.

3.2. Reading OIDs

The simplest way to get specific information from the camera is to issue an http request to read an OID value. For example, issuing the following http request:

```
http://<camera_ip>/get.oid?2.4
```

will return a page that has the value of OID 2.4 (the camera name) as the page contents.

3.3. Setting OIDs

When setting OID values the OID format contains additional information used to specify how the OID value is to be written. The format of the OID command is:

set.oid?*Oid*{*type*} {*style*}n.n.n = {*value*}

Where the following parameters must be defined:

type	The type of field that is being set. Use the value "T" for text entry fields.
style	Where the variable should be written. Specify "N" for writing to the camera's non-volatile memory, "R" for setting the run-time value only or "B" for both run-time and non-volatile memory.
value	The value to assign to the specified OID.

For example, to change the camera name to "IQcam" in the running system configuration use the http request:

```
http://<camera_ip>/set.oid?OidTR2.4="IQcam"
```

Note: The syntax of the set.oid URL is case sensitive.

If this request completes successfully the camera name will be changed to the string "IQcam" in the run-time system. The permanent value stored in non-volatile memory will not be changed.

In order to change permanent parameters using the set.oid URL the request must be authenticated using either basic or secure authentication. Basic authentication can be specified by including the username and password as part of the URL. If the privileged password has not been changed use the URL:

```
http://root:system@<camera_ip>/set.oid?OidTB2.4="IQcam"
```

This URL will permanently change the camera name to "IQcam" in both the runtime system and non-volatile memory.

Note: Internet Explorer does not allow basic authentication to be specified as part of a URL.

3.4. Efficiently setting multiple OID values

To efficiently set multiple OID values use the runtime option to change them all in the currently running system. When all OIDs in a functional group have been specified then write the single "Write to NVR" OID for the functional group. This will commit all the runtime values to the flash memory.

3.5. Commonly used OIDs

The following tables will show commonly used OIDs. Note that different camera models have varying functionality. Please see "OID supported matrix" in Section 3.5.12 for the list of OIDs that each camera family supports.

Note: For a complete list of available OIDs, request the URL http://<camera_ip>/oidtable.html from any IQinVision camera.

For more information on any of these OIDs see the IQinVision Reference Manual.

3.5.1. Audio support

Name	OID Value	Description
Audio input state	1.9.5	Enable/disable the audio input (microphone). Default is Disabled.
Audio input gain	1.9.1	Input amplification amount. Possible values are low, medium and high with the default value medium.
Audio output state	1.9.6	Enable/disable the audio output (speaker). Default is disabled.
Audio output gain	1.9.4	Speaker amplification amount. Possible values are low, medium and high with the default value medium.
Audio duplex mode	1.9.7	Determines whether audio is half or full duplex. Default is full.
Audio password	1.9.10	Password required to send audio to the camera. The username is "audio" and the default password is "sound". Note that these passwords are case sensitive.

3.5.2. Crop window

Name	OID Value	Description
Top edge	1.2.6.1	Top edge of crop window in pixels. Default is 0.
Width	1.2.6.2	Width of crop window in pixels. Default value is the width of the imager itself.
Height	1.2.6.3	Height of crop window in pixels. Default value is the height of the imager itself.
Left edge	1.2.6.4	Left edge of crop window in pixels. Default is 0.
Reset crop to maximum size	1.2.6.5	Writing this write only OID with the value "1" will reset the crop window back to factory default settings.

The crop window coordinate system uses the value (0,0) as the upper left corner of the image.

The default size for the crop window is the entire imager.

3.5.3. Flash disk

Name	OID Value	Description
Disk mounted	2.40.4.1.1	Read only OID returns the value "1" if a CF or SDHC media card has been successfully mounted by the camera.
Size of disk	2.40.4.1.2	Read only OID returns the size of the mounted media in megabytes.
Flash format	2.40.4.1.3	Read only OID returns the filesystem format of the mounted media. The only supported value is FAT32.
Free space	2.40.4.1.4	Read only OID returns the amount of free space in megabytes on the mounted filesystem.
Format disk	19.1	Writing this write only OID with the value "1" will format the current flash disk using the FAT32 file system format.

Before a CF or SDHC card can be recognized as a valid file system by the camera it must be formatted using the FAT32 file system. This can either be done using an external system or by writing the "Format Disk" OID.

Note: Formatting a disk will erase all the existing data without any warning.

3.5.4. General camera

Name	OID Value	Description
Camera name	2.4	Name of the camera as displayed in the web browser. Defaults to IQEYExxxxxx where xxxxxx is the last six characters of the MAC address.
Firmware version	2.5	Read only OID returns the camera version string in the format Version Vn.n/n(yymmdd).
Uptime	2.6	Read only OID returns the camera uptime as a text string.
Ethernet address	2.7	Read only OID returns the Ethernet (MAC) address of the camera in the format xx-xx-xx-xx-xx-xx.
Product name	2.8	Read only OID returns the product name, i.e. IQeye802.
Reboot camera	2.11	Writing this write only OID with the value "1" will reboot the camera.
Camera ID code	2.21	Read only OID returns the two character product family code.
Company Name	2.27	Read only OID returns the manufacturer name. Currently this value is IQinVision except for private labeled or OEM cameras.

3.5.5. H.264 control

Name	OID Value	Description
H.264 enable	1.17.7	Enable/Disable all H.264 functionality on IQA2xx series cameras. Default is enabled.
Audio enable	1.17.1.1	Enable/Disable audio support. Default is disabled.
Audio bitrate	1.17.1.2	Audio bit rate in Kbps. For the IQA2xx series cameras the default value is 16. For the IQ73xx series the value is fixed at 64 Kbits/second.
Video enable	1.17.2.1	Enable/Disable video support on IQA2xx cameras and is read only for IQ73xx series cameras. Default is enabled.
Video bitrate	1.17.2.2	Video bit rate in Kbps of the primary H.264 stream. The default value depends on the stream being requested. For CBR connections this is the CBR rate. For VBR connections the value specified is the average bit rate of what he camera will deliver.
Video deblocking	1.17.2.3	Enables or disables the in-loop deblocking filter and is read only for IQ73xx series cameras. Default is enabled.
Video frame rate	1.17.2.4	Video frame rate in frames per second. Default is 29.97 frames per second for IQA2xx series cameras and 30 frames per second for IQ73xx series cameras.
Video iframe	1.17.2.5	iFrame interval in milliseconds. Default value is 1000.
Video Mode	1.17.2.6	Specifies the size of the H.264 video stream. For the IQA2xx series cameras this value will always be VGA. For IQ73xx series cameras this value will return the mode specified by the administrator.
Video style	1.17.2.7	Specifies constant bit rate or variable bit rate. Options are CBR or VBR. Default is VBR.
Video preference	1.17.2.8	Specifies if the codec will drop quality to maintain frame rate (FRAMERATE) or drop frame rate to maintain quality (QUALITY). These options are primarily applicable if configured for a constant bit rate stream. Default is FRAMERATE.
RTSP listener port	1.17.3.1	Base IP listener port for RTSP connections. Default is 554.
H.264 factory default	1.17.4	Writing this write-only OID with the value “1” will return all H.264 parameters to their default values
H.264 save runtime to NVR	1.17.5	Writing this write-only OID with the value “1” will write all current H.264 runtime parameters to NVR.
H.264 installed	1.17.6	This read-only OID will return ENABLED if the camera is H.264 enabled and DISABLED if the camera is not capable of serving a H.264 stream.
Primary encoder width	1.17.8.1	This read-only OID will return the width in pixels of the primary H.264 stream.
Primary encoder height	1.17.8.2	This read-only OID will return the height in pixels of the primary H.264 stream.
Secondary encoder width	1.17.9.1	This read-only OID will return the width in pixels of the secondary H.264 stream.
Secondary encoder height	1.17.9.2	This read-only OID will return the height in pixels of the secondary H.264 stream.

3.5.6. H.264 RTSP stream control

Selected H.264 cameras have the ability to associate image parameters with streams. The OIDs in this table are available for stream "n" where the number of streams is determined by the camera model. The value with index 0 is a comma separated list of the current settings for all available streams.

Name	OID Value	Description
Available streams	1.17.2.18	This read-only OID returns the list of available streams.
Stream <n> codec	1.17.2.19.1.n	This set of OID values specifies what codec will be used for each stream.
Stream <n> resolution	1.17.2.19.2.n	This set of OID values specifies the resolution of each stream.
Stream <n> framerate	1.17.2.19.3.n	This set of OID values specifies the frame rate of each stream.
Stream <n> video style	1.17.2.19.4.n	This set of OID values specifies the video format of each stream. Options are "CBR" or "VBR".
Stream <n> bitrate	1.17.2.19.5.n	This set of OID values specifies the video bitrate in Kbits/second of each stream.
Stream <n> profile	1.17.2.19.6.n	This set of OID values specifies the H.264 profile of each stream. Options are "MAIN" or "BASELINE".
Stream <n> width	1.17.2.19.7.n	This set of OID values specifies the width in pixels of each stream.
Stream <n> height	1.17.2.19.8.n	This set of OID values specifies the height in pixels of each stream.
Stream <n> left edge	1.17.2.19.9.n	This set of OID values specifies the left edge of each stream.
Stream <n> top edge	1.17.2.19.10.n	This set of OID values specifies the upper edge of each stream.

3.5.7. Image control

Name	OID Value	Description
Gamma	1.2.1	Image gamma setting. Default value depends on the camera model.
Downsample	1.2.2	Specifies the factor that will be used to reduce the size of an image when a default image is requested. Values are powers of two, i.e. 1, 2, 4, 8.
Sharpness	1.2.3	Sharpening to be applied to images. Default depends on the camera model.
Image flip	1.2.4	Specifies how the image should be manipulated. Permissible values depend on the camera model. Default is no flip.
JPEG Quality	1.2.7	Specifies how much compression should be applied when creating JPEG images. Higher values have higher quality, i.e. the least compression. Different camera models have different ranges so see the individual camera documentation for the specific range.
Lighting Frequency	1.2.12.1	Lighting frequency to be used to compensate for fluorescent lighting beating. Valid values are 50 or 60. Default is 60.
Saturation	1.2.21	Color saturation target. Specifying higher saturation values will result in move vivid colors. Default value depends on the camera model.
Lightgrabber	1.2.30	Lightgrabber setting. Permissible values are 0, 1, 2, 4. Default value depends on the camera model.
Maximum frame rate	1.2.31	Maximum frame rate the camera will be allowed to deliver. Default is half of the maximum frame rate for each camera model.
Reset image parameters to factory defaults	1.2.14	Writing this write only OID with the value “1” will reset all image parameters back to their factory default values.

3.5.8. Image gain control

Name	OID Value	Description
Left edge main	1.2.8.1	Left edge of main gain window in pixels. Default is zero.
Width main	1.2.8.2	Width of main gain window in pixels. Default value is the width of the imager itself.
Height main	1.2.8.3	Height of main gain window in pixels. Default value is the height of the imager itself.
Top edge main	1.2.8.8	Top edge position of the main gain window in pixels. Default is 0.
Left edge	1.2.8.23.1.1.n	Left edge of gain window <n> in pixels.
Top edge	1.2.8.23.1.2.n	Top edge of gain window <n> in pixels.
Width	1.2.8.23.1.3.n	Width of gain window <n> in pixels.
Height	1.2.8.23.1.4.n	Height of gain window <n> in pixels.
Include	1.2.8.23.1.5.n	Specifies if window <n> is type include, meaning use the pixels in this window for gain calculations. Options are “ON” and “OFF”.
Exclude	1.2.8.23.1.6.n	Specifies if window <n> is type exclude, meaning do not use the pixels in this window for gain calculations. Options are “ON” and “OFF”.
Delete	1.2.8.23.1.7.n	Writing this write only OID with the value “1” will delete the window <n>.
Autogain target	1.2.8.4	Target brightness level of the image. Values range from 0.0 to 1.0 with higher values being brighter.
Gain hold value	1.2.8.5	TBD
Autogain state	1.2.8.6	TBD
Current gain value	1.2.8.7	This read only OID will return the current gain value. Values range from 0.0 to 1.0 with higher values being brighter.
Reset gain window to max size	1.2.8.9	Writing this write only OID with the value “1” will reset the image gain windows to defaults. All vs main?
Gain style	1.2.8.14	Determines how the camera adjusts the image brightness. Options are AVERAGE, CLIPAVERAGE, DARKDETECT and PEAKDETECT.
Gain shutter algorithm	1.2.8.12	Determines what algorithm will be used for imaging. Options are NORMAL, QUALITY, SPEED or a fixed shutter speed value.
Autogain adjust delay	1.2.8.22	Determines how fast the camera will react to changes in image brightness. Values range from 0.0 to 15.0 with lower values reacting faster.
Reset gain parameters to factory defaults	1.2.8.15	Writing this write only OID with the value “1” will reset all image gain parameters back to their factory default values.

3.5.9. Input/Output control

Name	OID Value	Description
LED activity	1.2.17	Determines what the camera LED will represent. Options are OFF, ACTIVITY and ON.
LED2 activity	2.45	Determines what the camera LED will represent. Options are ENABLED, DISABLED and TIMEOUT.
Input state	1.3.15	Defines how the input from an attached device will cause a trigger. Options are ONOPEN and ONCLOSE.
Simulate input trigger	1.3.10	Writing this write only OID with the value "1" will cause a trigger event just as if the input trigger activated.
Output relay state	1.3.5.1	Defines the normal (inactive) state of the output relay. Options are OPEN and CLOSED.
Output relay duration	1.3.5.2	Defines how long the output relay will be held in the active state when triggered. Values can range from 1 to 60 seconds.
Simulate output trigger	1.3.11	Writing this write only OID with the value "1" will active the output relay.
Last trigger was closed	1.3.17	This read only OID will return the time of the last trigger event in seconds since the epoch. (1/1/1970 GMT)
Activate output relay	17.4	TBD

3.5.10. Motion windows

Name	OID Value	Description
Left edge	1.11.1.1.n	Left edge of motion window <n> in pixels.
Top edge	1.11.1.2.n	Top edge of motion window <n> in pixels.
Width	1.11.1.3.n	Width of motion window <n> in pixels.
Height	1.11.1.4.n	Height of motion window <n> in pixels.
Include window	1.11.1.5.n	Specifies if window <n> is type include, meaning use this window for detecting motion. Options are "ON" and "OFF".
Exclude window	1.11.1.6.n	Specifies if window <n> is type exclude meaning do not detect motion in this area. Options are "ON" and "OFF"
Enable	1.11.1.7.n	Specifies if window <n> should be used. Options are "ENABLED" and "DISABLED"
Sensitivity	1.11.1.8.n	Specifies how sensitive window <n> should be for detecting motion. Values range from 0 to 254. The default value is 0 which means automatic sensitivity adjustment based on lighting conditions.
Percentage of window	1.11.1.9.n	Specifies how many of the pixels in window <n> must change for motion to be detected. Values range from 1% to 100%.
Remove all windows	1.11.4	Writing this write only OID will reset the motion windows to their factory default state.

Not all motion window OIDs are available for all camera models. See Section 3.6.10, "Motion Windows Support" for detailed information.

3.5.11. Overlay control

Name	OID Value	Description
Overlay state	1.2.9.1.8.n	Overlay <n> state.
Overlay text	1.2.9.1.9.n	Overlay <n> text string
Reset to Factory Default	1.2.9.3	Writing this write only OID with the value "1" will reset the overlay parameters to their factory default values.
Save to NVR	1.2.9.4	Writing this write only OID with the value "1" will save all current overlay values to NVR.

3.5.12. Virtual camera (VCAM)

Name	OID Value	Description
Window Name	1.16.1.1.n	Virtual camera <n> name
Frame rate	1.16.1.2.n	Frame rate for virtual camera <n>. Note that this value cannot be larger than the imager frame rate.
Left edge	1.16.1.3.n	Left edge of virtual camera source window <n> in pixels.
Top edge	1.16.1.4.n	Top edge of virtual camera source window <n> in pixels.
Width	1.16.1.5.n	Width of virtual camera source window <n> in pixels.
Height	1.16.1.6.n	Height of virtual camera source window <n> in pixels.
Destination width	1.16.1.7.n	Destination width of virtual camera <n> in pixels.
Destination height	1.16.1.8.n	Destination height of virtual camera <n> in pixels.
JPEG quality	1.16.1.9.n	JPEG compression value for virtual camera <n>. (not implemented)
Grayscale	1.16.1.10.n	Grayscale value for virtual camera <n>. (not implemented)
Overlay	1.16.1.11.n	Overlay text to be inserted into virtual camera <n>.
Set defaults	1.16.1.12.n	Set default values for virtual window <n>.
Save configuration to NVR	1.16.2	Save all configured VCAM windows to Non-Volatile flash memory.
Number of windows	1.16.3	Total number of configured VCAM windows.
Reset to Factory Default	1.16.4	Reset all VCAM configuration to factory default settings.
Window index list	1.16.5	List of all configured VCAM window indexes.

The Virtual Camera (VCAM) functionality is available on specific IQinVision camera models. See Section 3.6.12, "Virtual Camera (VCAM) Support" for detailed information.

3.6. OIDs supported by camera family

The following tables show which OIDs are supported by each camera family.

3.6.1. Audio support

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Audio input state	1.9.5						X *	X
Audio input gain	1.9.1						X *	X
Audio output state	1.9.6						X *	X
Audio output gain	1.9.4						X *	X
Audio duplex mode	1.9.7						X *	X
Audio password	1.9.10						X *	X

* Audio is supported on the IQ732, IQD3xx and IQM3xx families

3.6.2. Crop window

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Top edge	1.2.6.1	X	X	X	X	X		
Width	1.2.6.2	X	X	X	X	X		
Height	1.2.6.3	X	X	X	X	X		
Left edge	1.2.6.4	X	X	X	X	X		
Reset crop to maximum size	1.2.6.5	X	X	X	X	X		

3.6.3. Flash disk

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Disk mounted	2.40.4.1.1			X	X	X	X *	X
Size of disk	2.40.4.1.2			X	X	X	X *	X
Flash format	2.40.4.1.3			X	X	X	X *	X
Free space	2.40.4.1.4			X	X	X	X *	X
Format disk	19.1			X	X	X		X

* Flash disk supported on IQ732 only

3.6.4. General camera

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Camera name	2.4	X	X	X	X	X	X	X
Firmware version	2.5	X	X	X	X	X	X	X
Uptime	2.6	<X>	X	X	X	X	X	X
Ethernet address	2.7	X	X	X	X	X	X	X
Product name	2.8	X	X	X	X	X	X	X
Reboot camera	2.11	X	X	X	X	X	X	X
Camera ID code	2.21	X	X	X	X	X	X	X
Company Name	2.27	X	X	X	X	X	X	X

3.6.5. H.264 control

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
H.264 enable	1.17.7					X		
Audio enable	1.17.1.1					X		
Audio bitrate	1.17.1.2					X	X	X
Video enable	1.17.2.1					X	X	X
Video bitrate	1.17.2.2					X	X	X
Video deblocking	1.17.2.3					X		
Video frame rate	1.17.2.4					X	X *	X*
Video iframe	1.17.2.5					X		
Video Mode	1.17.2.6					X	X	X
Video style	1.17.2.7					X	X	X
Video preference	1.17.2.8					X		
Motion SEI enable	1.17.2.15						X	X
RTSP listener port	1.17.3.1					X	X	X
H.264 factory default	1.17.4					X	X	X
H.264 save runtime to NVR	1.17.5					X		
H.264 installed	1.17.6					X	X	X
Primary encoder width	1.17.8.1						X	X
Primary encoder height	1.17.8.2						X	X
Secondary encoder width	1.17.9.1						X	X
Secondary encoder height	1.17.9.2						X	X

* The OID parameter applies to both primary and secondary H.264 streams.

3.6.6. H.264 RTSP stream control

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Available streams	1.17.2.18							X
Stream <n> codec	1.17.2.19.1.n							X
Stream <n> resolution	1.17.2.19.2.n							X
Stream <n> framerate	1.17.2.19.3.n							X
Stream <n> video style	1.17.2.19.4.n							X
Stream <n> bitrate	1.17.2.19.5.n							X
Stream <n> profile	1.17.2.19.6.n							X
Stream <n> width	1.17.2.19.7.n							X
Stream <n> height	1.17.2.19.8.n							X
Stream <n> left edge	1.17.2.19.9.n							X
Stream <n> top edge	1.17.2.19.10.n							X

3.6.7. Image control

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Gamma	1.2.1	X	X	X	X	X	X	X
Downsample	1.2.2		X	X	X	X		
Sharpness	1.2.3	X	X	X	X	X	X	X
Image flip	1.2.4		X	X	X	X	X	X
JPEG Quality	1.2.7	X	X	X	X	X	X	X
Lighting Frequency	1.2.12.1	X	X	X	X	X	X	X
Saturation	1.2.21	X	X	X	X	X	X	X
Lightgrabber	1.2.30		X	X	X	X	X	X
Maximum frame rate	1.2.31	X	X	X	X	X	<X>	<X>
Reset image parameters to factory defaults	1.2.14	X	X	X	X	X	X	X

3.6.8. Image gain control

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Left edge main	1.2.8.1		X	X	X	X	X	X
Width main	1.2.8.2		X	X	X	X	X	X
Height main	1.2.8.3		X	X	X	X	X	X
Top edge main	1.2.8.8		X	X	X	X	X	X
Left edge	1.2.8.23.1.1.n			X	X	X	X	X
Top edge	1.2.8.23.1.2.n			X	X	X	X	X
Width	1.2.8.23.1.3.n			X	X	X	X	X
Height	1.2.8.23.1.4.n			X	X	X	X	X
Include	1.2.8.23.1.5.n			X	X	X	X	X
Exclude	1.2.8.23.1.6.n			X	X	X	X	X
Delete	1.2.8.23.1.7.n			X	X	X	X	X
Autogain target	1.2.8.4	X	X	X	X	X	X	X
Gain hold value	1.2.8.5		X	X	X	X		
Autogain state	1.2.8.6		X	X	X	X		
Current gain value	1.2.8.7		X	X	X	X	X	X
Reset gain to max size	1.2.8.9		X	X	X	X	X	X
Gain style	1.2.8.14		X	X	X	X	X	X
Gain shutter algorithm	1.2.8.12		X	X	X	X	X	X
Autogain adjust delay	1.2.8.22		X	X	X	X		
Reset gain parameters to factory defaults	1.2.8.15		X	X	X	X		

3.6.9. Input/Output control

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
LED activity	1.2.17	X	X	X	X	X	X *	
LED2 activity	2.45						X *	X
Input state	1.3.15			X	X	X	X	X
Simulate input trigger	1.3.10			X	X	X	X	X
Output relay state	1.3.5.1			X	X	X	X	X
Output relay duration	1.3.5.2			X	X	X	X	X
Simulate output trigger	1.3.11			X	X	X	X	X
Last trigger event time	1.3.20			X	X	X	X	X
Activate output relay	17.4			X	X	X	<X>	X

* The IQM3xx and IQD3xx families support OID 1.2.17. The IQ73xx and IQ83xx families support 2.45.

3.6.10. Motion windows

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Left edge	1.11.1.1.index		X	X	X	X	X	X
Top edge	1.11.1.2.index		X	X	X	X	X	X
Width	1.11.1.3.index		X	X	X	X	X	X
Height	1.11.1.4.index		X	X	X	X	X	X
Include window	1.11.1.5.index		X	X	X	X		
Exclude window	1.11.1.6.index		X	X	X	X		
Enable	1.11.1.7.index		X	X	X	X	X	X
Sensitivity	1.11.1.8.index	X	X	X	X	X	X	X
Percentage of window	1.11.1.9.index	X	X	X	X	X	X	X
Remove all windows	1.11.4		X	X	X	X	X	X

3.6.11. Overlay control

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Overlay state	1.2.9.1.8.index		X	X	X	X	X	X
Overlay text	1.2.9.1.9.index		X	X	X	X	X	X
Reset overlay to Factory Default	1.2.9.3		X	X	X	X	X	X
Save to NVR	1.2.9.4		X	X	X	X		

3.6.12. Virtual camera (VCAM)

Name	OID Value	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Window Name	1.16.1.1.n			X	X	X		
Frame rate	1.16.1.2.n			X	X	X		
Left edge	1.16.1.3.n			X	X	X		
Top edge	1.16.1.4.n			X	X	X		
Width	1.16.1.5.n			X	X	X		
Height	1.16.1.6.n			X	X	X		
Destination width	1.16.1.7.n			X	X	X		
Destination height	1.16.1.8.n			X	X	X		
JPEG quality	1.16.1.9.n			X	X	X		
Grayscale	1.16.1.10.n			X	X	X		
Overlay	1.16.1.11.n			X	X	X		
Set defaults	1.16.1.12.n			X	X	X		
Save configuration to NVR	1.16.2			X	X	X		
Number of windows	1.16.3			X	X	X		
Reset to Factory Default	1.16.4			X	X	X		
Window index list	1.16.5			X	X	X		

4. Image Data Formats

IQinVision cameras can deliver image data in a number of different formats. The basic structure of these formats will be documented in this chapter. Please see the following chapters for specific information on how to request data.

Note: Not all camera models can return all types of image data. Please see the specific camera family information for a list of the capabilities of each model.

4.1. Image data supported by camera family

The following table shows which image formats are supported by each camera family.

Modifier	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
Single JPEG image	X	X	X	X	X	X	X
JPEG serverpush stream	X	X	X	X	X	X *	X*
PGM grayscale			X	X	X		
PGM grayscale serverpush stream			X	X	X		
JPEG serverpush stream with embedded motion data	X	X	X	X	X	X *	X*
Serverpush stream with JPEG images only when motion detected		X	X	X	X		
Serverpush stream with only embedded motion metadata		X	X	X	X		
Standalone motion data stream		X	X	X	X		
Motion bitmap			X	X	X		
H.264 stream over Unicast UDP					X	X	X
H.264 stream over Multicast UDP						X	X
H.264 stream over TCP						X	X
H.264 stream over HTTP via TCP						X	X

* JPEG data availability depends on the H.264 mode or the streams configuration.

4.2. URI modifiers supported by camera family

The following table shows which URI modifiers are supported by each camera family.

Modifier	IQ04xx IQD4xx IQ54xx	IQ51x	IQ70x IQ75x IQ8xx	IQA1xx	IQA2xx	IQ73xx IQ83xx IQD3xx IQM3xx	IQA3xx
snap=spushn	X	X	X	X	X	X	X
snap=spushn&pragma=motion	X	X	X	X	X	X	X
snap=spushn&pragma=trigger		X	X	X	X		
snap=spushn&pramga=motion-only		X	X	X	X		
snap=spushn&pragma=motion &noimage		X	X	X	X		
ds		X	X	X	X		
wl			X	X	X		
wt			X	X	X		
wrw			X	X	X		
wrh			X	X	X		
wdw			X	X	X		
wdh			X	X	X		
mm					X	X	X
res						X *	X

* The IQ73xx/IQ83xx/IQD3xx/IQM3xx families only support “low” and “high” as parameters.

5. Single JPEG Image

All IQinVision cameras can deliver single JPEG images. The JPEG image format is defined by the Joint Photographic Experts Group and can be seen in ISO document 10918.

IQinVision cameras include a default comment in the JPEG header to identify the camera which created the image. This comment consists of the six byte MAC address followed by a single byte which is the camera product code.

5.1. Requesting a single JPEG image

The simplest method to get a single frame of data is to use an http request to get an image. The basic URL used to request a single JPEG image is:

```
http://<camera_ip>/now.jpg
```

This http request can be issued in a number of ways:

- Using a browser.
- Using WinInet calls on a Win32 platform. Please see http://msdn.microsoft.com and search for WinInet for an overview of the provided functionality and interfaces.
- Using PERL scripting. For talking over the web with Perl, the defacto standard is the LWP set of modules. (LWP stands for "Library for WWW in Perl") Please see http://www.cpan.org/ for LWP documentation and examples.
- Using direct TCP connection requests.

Regardless of the mechanism used to issue the http request, the camera will return the specified image in the format requested.

6. Serverpush Stream

Server push is a method for sending updating data to a web page. The camera sends streaming images as server push streams of JPEG images separated by HTML tags using multipart MIME encoding. The camera will continue sending image data as fast as it is allowed until the HTTP connection is closed.

6.1. Serverpush stream format

The camera will deliver a server push image stream in response to requests for the image file:

```
http://<camera_ip>/now.jpg?snap=spush
```

Image modifiers may be added as described in the next section.

The data returned by the camera is formatted as a series of JPEG images separated by HTML tags. The HTTP response from camera will look like:

```
HTTP/1.0 200 OK
Cache-Control: no-cache
Content-Type: multipart/x-mixed-replace; boundary=--ImageSeparator

--ImageSeparator
Content-Type: image/jpeg
Content-Length: <nnnnn>

<JPEG Image>

--ImageSeparator
Content-Type: image/jpeg
Content-Length: <nnnnn>

<JPEG Image>

--ImageSeparator
Content-Type: image/jpeg
Content-Length: <nnnnn>

<JPEG Image>

--ImageSeparator
```

The camera will continue sending image data as fast as it is allowed until the http (TCP) connection is closed.

6.2. Requesting a serverpush stream

URI modifiers can be added to modify the image that is retrieved by the HTTP request, without having to change the camera's configuration. For example, to retrieve the current image downsampled by a factor of 2, use the following URL:

```
http://<camera_ip>/now.jpg?ds=2
```

Note: URLs are specified in the form now.jpg?x=y&a=b&c=d.

The modifiers supported by each camera family will documented in the product specific chapters.

6.2.1. Serverpush modifiers

The following table explains serverpush modifiers.

Modifier	X value	Effect
snap=X	Spush	Uses server push connection
	Spushn	Uses server push, but updates it at most every n seconds (n can be a floating point) Spush with a value of 0 will send images as fast as the camera can acquire them. Spush with a value of 0.2 will send five images per second.

For example, to request a serverpush stream downsampled by four at a frame rate of 4 frames per second use the following URL:

```
http://<camera_ip>/now.jpg?snap=spush0.25?ds=4
```

6.2.2. Image control modifiers

Image control.

Modifier	X value	Effect
ds=n		Downsamples image by the value of n (1, 2, 4, 8)

6.2.3. Source/Destination resolution

Some cameras support the ability to request a "destination resolution" rather than just an integer downsample value. The camera will format the source video to fit into the desired destination resolution. This insulates the video recording system from having to know the specific resolution of the camera and just request the desired resolution. This could be quite useful for things like multi-view displays.

Modifier	Effect
wl=n	window absolute left (lowercase "L" not the number "1")
wt=n	window absolute top
wrw=n	window absolute raw width
wrh=n	window absolute raw height
wdw=n	window destination width
wdh=n	Windows destination height. If only one of the "wdw" or "wdh" parameters is specified a 1:1 aspect ratio will be assumed.

Note: Absolute coordinates have (0,0) at the top left corner of the image.

For an example, to ask for a stream of 320 by 240 pixels, 10 pixels to the right and 20 pixels down from the upper left corner of the full image, use the URL:

```
http://<camera_ip>/now.jpg?snap=spush&wl=10&wt=20&wrw=320&wrh=240
```

This request will deliver the 320x240 image as a separate serverpush stream, completely independent of any other streams of data on the camera.

The camera can also do scaling to take a source image and "fit it into" a destination region. For example, to ask for a scaled-up version of a stream of 320 by 240 pixels, 10 pixels to the right and 20 pixels down from the upper left corner of the full image, use the URL:

```
http://<camera_ip>/now.jpg?snap=spush&wl=10&wt=20&wrw=320&wrh=240&wdw=640
```

This will resample the image using bicubic interpolation and deliver it as a 640x480 stream

6.3. Serverpush stream with embedded motion data

The simplest mechanism to get motion data from the camera is to request embedded motion meta data in the serverpush stream. To enable the status information, the serverpush URL request should include the "pragma=motion" string, as shown in this sample URL:

```
http://<camera_IP>/now.jpg?snap=spush&pragma=motion
```

This pragma will return the raw motion detection from each frame. As long as motion continues to occur, this option will continue showing the motion indication for each frame of video data.

6.4. Serverpush stream with embedded trigger data

If requested, the camera can include both motion and trigger status information in the image data it serves in a serverpush stream. To enable the status information, the serverpush URL request should include the "pragma=trigger" string, as shown in this sample URL:

```
http://<camera_IP>/now.jpg?snap=spush&pragma=trigger
```

Each HTML image header will then contain a line of the form "Pragma: trigger=<reason>," where reason can be one of the following strings:

"none" - no trigger event occurred.
"input" - the I/O relay was triggered.
"timer" - a periodic (time-based) trigger occurred.
"motion" - the motion detection software triggered the event. The pragma line will also include the number(s) of the motion detection windows that detected the motion. Note that this will only show the motion pragma in the "first" frame of the motion sequence. The trigger will be re-enabled when ????
"test" - a one-time trigger occurred. A user ran either the `TEST TRIGGER` command or the equivalent OID.

For example, a sample image stream that detected motion in motion windows 2 and 3 might contain:

```
Pragma: trigger=none
Pragma: trigger=none
Pragma: trigger=motion 2 3
Pragma: trigger=none
```

The first image in the serverpush stream will always have `pragma: trigger=none.`

6.5. Motion only serverpush stream

The camera can deliver a stream of images only when motion is detected. Request this stream using the standard now.jpg?snap=spush URL and include the URI modifier ?motiononly.

```
http://<camera_IP>/now.jpg?snap=spush&pragma=motiononly
```

6.6. Serverpush stream with no image data

The camera can deliver a serverpush stream with motion data and no image frames. Request this stream using the standard now.jpg?snap=spush. URL and include the URI modifiers ?pragma=motion and ?noimage.

```
http://<camera_IP>/now.jpg?snap=spush&pragma=motion&noimage
```

This will deliver a HTTP stream with the following format:

```
HTTP/1.0 200 OK
Cache-Control: no-cache
Content-Type: multipart/x-mixed-replace; boundary=--ImageSeparator

--ImageSeparator
Content-Type: image/jpeg
Content-Length: 0
Pragma: trigger=none

--ImageSeparator
Content-Type: image/jpeg
Content-Length: 0
Pragma: trigger=motion 1 2

--ImageSeparator
Content-Type: image/jpeg
Content-Length: 0
Pragma: trigger=motion 1
```

7. PGM Image Data

The IQeye 7xx and 8xx series cameras will have the ability to provide eight-bit monochrome data extracted from the Y plane of YCbCr frames.

7.1. PGM grayscale data

To request the monochrome (Y plane) data as a single frame use the URL:

```
http://<camera_ip>/now.pgm?ds=<size>
```

Where size is one of VGA(640x480), QVGA (320x240), QQVGA (160x120), or QQQVGA (80x60). This will return a standard Portable Gray Map (PGM) bitmap in the format:

```
P5
#<sequence number>
<image width>
<image height>
<max grayscale value>
nnn nnn nnn nnn nnn nnn nnn nnn nnn <width values> nnn nnn nnn

(grayscale data for the entire requested image)

nnn nnn nnn nnn nnn nnn nnn nnn nnn <width values> nnn nnn nnn
```

The sequence number is an incrementing counter that can be used to identify this particular image.

7.2. PGM serverpush stream

To request a stream of PGM images, use the URL:

```
http://<camera_ip>/now.pgm?snap=spush?ds=<size>
```

Where size is one of VGA(640x480), QVGA (320x240), QQVGA (160x120), or QQQVGA (80x60).

This results in an HTTP stream that is very similar to a MJPEG serverpush stream, but the content type is x-portable-graymap.

```
HTTP/1.0 200 OK
Cache-Control: no-cache
Content-Type: multipart/x-mixed-replace; boundary=--ImageSeparator

--ImageSeparator
Content-Type: image/x-portable-graymap
Content-Length: <nnnn>
<PGM data>

--ImageSeparator
Content-Type: image/x-portable-graymap
Content-Length: <nnnn>
<PGM data>

--ImageSeparator
```

8. Standalone Motion Data

To request motion data without using a standard serverpush stream you must first enable motion detection on the camera either using the camera web pages or via OIDs. You then request the URL:

```
http://<camera_IP>/now.motiondata
```

This will return a stream of data of the format:

```
HTTP/1.0 200 OK
Cache-Control: no-cache
Content-Type: multipart/x-mixed-replace; boundary=--iqeyemd

--iqeyemd
Content-Type: text/plain

window=1;level=2;threshold=100
window=2;level=0;threshold=100
window=3;level=1;threshold=100
--iqeyemd
Content-Type: text/plain

window=1;level=0;threshold=100
window=2;level=0;threshold=100
window=3;level=0;threshold=100

(repeated as many times as the camera delivers frames in 200 milliseconds.  Note that
this could be zero times if the camera is using a very long exposure time in low light
conditions.)

(200 millisecond delay here)

--iqeyemd
Content-Type: text/plain

window=1;level=10;threshold=100
window=2;level=0;threshold=100
window=3;level=0;threshold=100
--iqeyemd
Content-Type: text/plain

window=1;level=0;threshold=100
window=2;level=10;threshold=100
window=3;level=17;threshold=100

(repeated as many times as the camera delivers frames in 200 milliseconds)

(200 millisecond delay here)

--iqeyemd
```

Information about the motion levels is reported every 200 milliseconds. The actual number of sets of window motion data in each 200 millisecond block is dependent on the frame rate at which the camera is capturing images.

Note: Motion detection must be enabled on the camera to use this type of motion data.

9. Bitmap Motion Map

The IQeye7xx and IQeye8xx series cameras can provide a "motion map" that shows where in the source image motion has been detected. To request a stream of PGM motion detection images, use the URL:

```
http://<camera_ip>/now.mmap?snap=spush
```

This results in an HTTP stream that is very similar to a MJPEG serverpush stream, but the content type is x-portable-bitmap. The bitmap size is always imager_cropped_width/8 by imager_cropped_height/8. Note that this is one bit per "motion detection block," i.e. one bit per 8x8 pixel block.

For example, for a 2 Megapixel sensor (1600x1200) the resulting motion bitmap will be 200x150 pixels. Since this is one bit per pixel, the actual data in the file will be 25 bytes (200 bits/8 bits per byte) by 150 rows, or 3750 image bytes.

The PBM bitmap format is:

```
P4
#<sequence number>
<image width>
<image height>
nnn nnn nnn nnn nnn nnn <width/8 values> nnn nnn nnn

(bitmap data for the entire requested image)

nnn nnn nnn nnn nnn nnn <width/8 values> nnn nnn nnn
```

10. H.264 Streams

Specific cameras from IQinVision are capable of providing H.264 RTP/RTSP image and audio streams. The requests are always made over TCP using a RTSP connection and the specifics of the content are negotiated as defined in RFC 3984, "RTP Payload Format for H.264 Video." These streams can be delivered using a number of different transport protocols and are documented in the following sections.

The IQA2xx (Legacy Alliance) camera models can provide a single H.264 stream at VGA resolution. The only transport method supported is H.264 via unicast UDP. It can also provide AAC encoded audio data from the camera to the host. Full duplex audio is not supported.

All other IQinVision H.264 camera models can be configured to operate in either mode or stream based operation. The following documentation will describe how to configure the H.264 encoders and how to request image data from these cameras.

10.1. Requesting H.264 image data

There are several different underlying protocols that can be used when requesting an H.264 stream using RTSP/RTP. Using a TCP based connection will result in a more reliable data stream, especially over Wide Area Network (WAN) connections, but places additional overhead on the host system. Using UDP is most efficient, but lost packets are not handled and images can be corrupted.

When configuring network firewalls, the following ports must be opened for H.264 connections.

- H.264 via unicast UDP: TCP port 554 is used for the RTSP control connection. The data is transmitted via UDP and the actual UDP port is negotiated. IQinVision cameras start the negotiation on UDP port 6970 for audio and 6972 for video. If the requested UDP port is busy it is incremented by one and requesting until the UDP port gets successfully bound.
- H.264 via multicast UDP: The port is configured by the user and the switches/routers must be configured to support the selected multicast group.
- H.264 over TCP: The RTSP and data connections are transmitted on TCP port 554.
- H.264 via TCP over HTTP: The RTSP and data connections are transmitted over TCP port 80.

10.1.1. Unicast/Multicast UDP or TCP stream

To request an H.264 stream via Unicast UDP, Multicast UDP or Unicast TCP, request a RTSP connection using the camera base IP address:

```
rtsp://<camera_ip>/now.mp4
```

Once the RTSP connection has been established the specific transport mechanism and H.264 request must then be negotiated using the protocol established in RFC 3984.

10.1.2. Unicast TCP stream tunneled over HTTP

To request an H.264 stream via Unicast TCP tunneled over HTTP request a http connection using the URL:

```
http://<camera_ip>/rtsp/now.mp4
```

Once the HTTP connection has been established the specific transport mechanism and H.264 request must then be negotiated using the protocol established in RFC 3984.

10.2. H.264 mode based operation

All current H.264 cameras support mode based operation. When using this style of interface an administrator can change the mode using the configuration web pages or an NVR platform can change the mode using OID values.

The mode specifies both the resolution and the codec (H.264 or MJPEG) configured for the primary and secondary image streams. The following tables show the image data that the cameras will deliver in each mode.

10.2.1. IQ732/IQ832/IQD3xx/IQM3xx modes of operation

Mode	Description	IQ73xx IQ83xx	IQM32x	IQD31x IQM31x	IQD30x IQM30x
Default (boot mode)		Mode0	Mode2	Mode2	Mode9
mode0	MJPEG (1280x720) Analog out, NTSC or PAL, Letterboxed full FOV	X			
mode1	H.264 1080p (1920x1080), CBR MJPEG (720x480)	X	X		
mode2	H.264 720p (1280x720), CBR or VBR MJPEG (1280x720)	X	X	X	
mode3 **	H.264 720p (1280x720), CBR or VBR H.264 480p (720x480), CBR or VBR MJPEG (720x480), approximately 1 fps	X	X	X	
mode4	H.264 720p (1280x720), CBR or VBR Analog output, NTSC or PAL, letterboxed full FOV MJPEG (720x360) if configured for NTSC MJPEG (720x432) if configured for PAL	X			
mode5	MJPEG (1920x1080)	X	X		
mode6	H.264 720p (1280x720), CBR or VBR MJPEG (720x480)	X	X	X	
mode7 **	H.264 720p (1280x720), CBR or VBR H.264 SIF (352x240), CBR or VBR MJPEG (352x240), approximately 1 fps	X	X	X	
mode8	H.264 720p (1280x720), CBR or VBR MJPEG (352x240)	X	X	X	
mode9	H.264 480p (720x480), CBR or VBR MJPEG (720x480)	X	X	X	X
mode10 **	H.264 480p (720x480), CBR or VBR H.264 SIF (352x240), CBR or VBR MJPEG (352x240), approximately 1 fps	X	X	X	X
mode11	H.264 480p (720x480), CBR or VBR MJPEG (352x240)	X	X	X	X

Note: The maximum frame rate for each mode is determined by the specifications specific camera model.

*** Both H.264 connections must use the same style, i.e. both must be CBR or both must be VBR.*

10.2.2. IQA3xx modes of operation

Mode	Description	IQA35N	IQA33N	IQA32N	IQA31N	IQA30N
IQA35 boot mode mode0	MJPEG, 5MP 4:3 (2592x1944), 5 fps Analog out, NTSC or PAL, full FOV	X				
IQA33N boot mode mode0	MJPEG, 3.6MP 16:9 (2560x1440), 8 fps Analog out, NTSC or PAL, full FOV		X			
IQA32N boot mode mode0	H.264 720p (1280x720), CBR or VBR MJPEG (1280x720) Analog out, NTSC or PAL, Letterboxed full FOV			X		
IQA31 boot mode mode0	H.264 720p (1280x720), CBR or VBR MJPEG (1280x720) Analog out, NTSC or PAL, Letterboxed full FOV				X	
IQA30 boot mode mode0	H.264 480p (720x480), CBR or VBR MJPEG (720x480) Analog out, NTSC or PAL, Letterboxed full FOV					X
mode1	H.264, 1080p (1920x1080), CBR MJPEG (720x480)	X	X	X		
mode2	H.264, 720p (1280x720), CBR or VBR MJPEG (1280x720), up to 30 fps	X	X	X	X	
mode3 **	H.264, 720p (1280x720), CBR or VBR H.264, 480p (720x480), CBR or VBR MJPEG (720x480), approximately 1 fps	X	X	X	X	
mode4	H.264, 720p (1280x720), CBR or VBR Analog output, NTSC or PAL, letterboxed full FOV MJPEG (720x360) configured for NTSC MJPEG (720x432) configured for PAL	X	X	X	X	
mode5	MJPEG (1920x1080)	X	X	X		
mode6	H.264, 720p (1280x720), CBR or VBR MJPEG, color (720x480)	X	X	X	X	
mode7 **	H.264, 720p (1280x720), CBR or VBR H.264, SIF (352x240), CBR or VBR MJPEG (352x240), approximately 1 fps	X	X	X	X	
mode8	H.264, 720p (1280x720), CBR or VBR MJPEG (352x240)	X	X	X	X	
mode9	H.264, 480p (720x480), CBR or VBR MJPEG (720x480)	X	X	X	X	X
mode10 **	H.264, 480p (720x480), CBR or VBR H.264, SIF (352x240), CBR or VBR MJPEG (352x240), approximately 1 fps	X	X	X	X	X
mode11	H.264, 480p (720x480), CBR or VBR MJPEG (352x240)	X	X	X	X	X
mode12	H.264, 5MP, 4:3 (2592x1944), CBR or VBR MJPEG (640x480)	X				

Mode	Description	IQA35N	IQA33N	IQA32N	IQA31N	IQA30N
mode13	H.264, 3.6MP, 16:9 (2560x1440), CBR or VBR MJPEG (720x480)	X	X			
mode14	H.264, 3MP , 4:3 (2048x1536), CBR or VBR MJPEG (640x480)	X	X			
mode15	H.264, 2MP, 4:3 (1600x1200), CBR or VBR MJPEG (640x480)	X	X	X		
mode16	H.264, 1MP, 4:3 (1280x960), CBR or VBR MJPEG (640x480)	X	X	X	X	
mode17	MJPEG (2592x1944) MJPEG (640x480)	X				
mode18	MJPEG (2560x1440) MJPEG (720x480)	X				
mode19	MJPEG (2048x1536) MJPEG (640x480)	X	X			
mode20	MJPEG (1600x1200) MJPEG (640x480)	X	X	X		
mode21	MJPEG (1280x960) MJPEG (640x480)	X	X	X	X	

** Both H.264 connections must use the same style, i.e. both must be CBR or both must be VBR.

When the camera has a factory default configuration it will boot and operate in the mode specified in the "boot mode" row.

For cameras with analog output an analog service monitor can be used to by an installer to set the camera's field of view and focus when the camera is in a mode that supports "Analog out."

10.3. H.264 stream based operation

The IQA3xx camera family supports stream based configuration. Using streams does not require configuring a specific mode of operation.

The OID values defined in Section 3.5.6, "H.264 RTSP stream control" can be used to configure the encoder, resolution and framerate parameters for camera streams. For example, to configure two streams with the first stream VBR H.264 encoding with a bit rate of 4000 KBits, 3 megapixel resolution and 5 frames per second and the second stream MJPEG, QVGA resolution, 5 frames per second set the OID values:

Name	OID Value	Value
Stream 1 codec	1.17.2.19.1.1	H264
Stream 1 resolution	1.17.2.19.2.1	3MP
Stream 1 framerate	1.17.2.19.3.1	5
Stream 1 bitrate mode	1.17.2.19.4.1	vbr
Stream 1 bit rate	1.17.2.19.5.1	4000
Stream 2 codec	1.17.2.19.1.2	jpeg
Stream 2 resolution	1.17.2.19.2.2	qvga
Stream 2 framerate	1.17.2.19.3.2	5

Always configure the streams in order starting with stream 1. Set the primary stream, i.e. stream 1, to the largest resolution and highest frame rate required by the user application. This stream will have the highest priority. The resources available after this stream has been configured determine the configurations available for the secondary, tertiary and quaternary streams.

To determine the number of streams that the camera supports read OID 1.17.2.18, List of available streams.

To request streams using RTSP use the following URL values.

Stream	URL
Stream 1	rtsp://<camera_ip>/stream1
Stream 2	rtsp://<camera_ip>/stream2
Stream 3	rtsp://<camera_ip>/stream3
Stream 4	rtsp://<camera_ip>/stream4

10.4. H.264 request URI modifiers

The following modifiers can be used when requesting H.264 streams.

Modifier	X value	Effect
mm=x	a	Deliver only audio data
	v	Deliver only video data
res=x	high	If the configured mode allows it, deliver the higher resolution H.264 stream
	low	If the configured mode allows it deliver the lower resolution H.264 stream
	5MP or 2592x1944	Deliver a 5MP resolution H.264 stream
	3MPwide or 2560x1440	Deliver a 3 MPwide (16:9) resolution H.264 stream
	3MP or 2048x1536	Deliver a 3 MP (4:3) resolution H.264 stream
	1080p or 1920x1080	Deliver a 1080p resolution H.264 stream
	2MP or 1600x1200	Deliver a 2 MP resolution H.264 stream
	1.2MP or 1280x960	Deliver a 1.2 (4:3) resolution H.264 stream
	720p or 1280x720	Deliver a 720p resolution H.264 stream
	480p or 720x480	Deliver a 480p resolution H.264 stream
	vga or 640x480	Deliver a VGA resolution H.264 stream
	sif or 352x240	Deliver a SIF resolution H.264 stream
	qvga or 320x240	Deliver a QVGA resolution H.264 stream

* The IQ73xx/IQ83xx/IQD3xx/IQM3xx families only support “low” and “high” as parameters.

These modifiers can be used to specify the resolution of the H.264 stream. Note that RTSP still needs to be used to negotiate the underlying transport protocol parameters. For example, to request a TCP based Unicast H.264 video only stream at a resolution of 1080p use the URL:

```
rtsp://<camera_ip>/now.mp4?res=1080p&mm=v
```

To request a H.264 stream tunneled over HTTP with 2MP resolution and both audio and video use the URL:

```
http://<camera_ip>/rtsp/now.mp4?res=1600x1200
```

10.5. H.264 stream with motion/time SEI data

IQinVision has added additional information into H.264 streams as a User Data Unregistered Supplemental Enhancement Information (SEI) messages. For additional information on SEI messages, refer to Annex D of ITU-T H.264.

Note: IQinVision defined SEI messages are disabled by default and must be enabled by setting oid 1.17.2.15 to 1.

The SEI's are found in the first RTP packet of each frame and have the following format.

```
nal_length [2 bytes]
nal_code [1 byte]
sei_payload_type [1 byte if type is less than 256 bytes]
sei_payload_length [1 byte if length is less than 256 bytes]
uuid [16 bytes]
user_data [user_data_len]
rbsp_trailing_bits [1 byte]
```

The user data may contain multiple messages concatenated together. Each message has a type, length and associated data bytes with the following format.

```
msg_type [1 byte]
msg_length [2 bytes]
msg_data [msg_length bytes]
```

For example, a frame with no motion would have an SEI with the format.

```
nal_length = 0x<nnnn> (length depends on data)
nal_code = 0x06 (SEI)
sei_payload_type = 0x05 (User data)
sei_payload_length = 0x37
uuid = 0x<nnnn> (16 bytes, camera specific)
msg_type = 0x01 (Pragma trigger message)
msg_length = 0x0014
msg_data = "Pragma: trigger=none"
msg_type = 0x02 (Timestamp message)
msg_length = 0x000d
msg_data = "1285697899672" (milliseconds since epoch)
rbsp_trailer = 0x80
```

A frame where motion had been detected in zones 1 and 3 will include the motion zone identifiers in the pragma text.

```
nal_length = 0x<nnnn> (length depends on data)
nal_code = 0x06 (SEI)
sei_payload_type = 0x05 (User data)
sei_payload_length = 0x3d
uuid = 0x<nnnn> (16 bytes, camera specific)
msg_type = 0x01 Pragma trigger message)
msg_length = 0x001a
msg_data = "Pragma: trigger=motion 1 3"
msg_type = 0x02 (Timestamp message)
msg_length = 0x000d
msg_data = "1285697899840" (milliseconds since epoch)
rbsp_trailer = 0x80
```

The motion detection may occur and the pragma information be inserted in a frame 0, 1 or 2 frames before the actual image data reflects motion. In other words, the SEI information may be considered a pre-trigger, but the number of pre-triggered frames is not deterministic.

11. Two-way Audio Support

Specific IQinVision cameras are capable of receiving digital audio data via an Ethernet TCP connection and playing it via the sound output device. The setup, data format and transport mechanism are documented in the following sections.

11.1. Camera setup

To setup the camera to play audio data the output device must first be enabled by configuring the Audio Output State, OID 1.9.6, to "enabled". This turns on the audio output hardware.

Next the Audio Output Gain, OID 1.9.4 can be configured. The default gain value is "medium" which is adequate for most installations. Please check the actual audio levels in the installation environment to decide how to configure this value.

11.2. Sending audio data

IQinVision cameras support playing mono, ulaw encoded, 16 bit PCM audio data sampled at 8 KHz.

Audio is sent to the camera by doing an HTTP POST to audiorx.cgi. Basic authentication credentials need to be supplied. The default audio username is "audio" and the default password is "sound". Note that the username cannot be changed, but the administrator can change the audio password using OID 1.9.10.

The audio data must be delivered to the camera at a real-time rate of 8 KHz. There is a four second audio buffer in the camera, so each individual HTTP POST should only contain several seconds of data. The host sending the audio data must manage the transmission of the data so that the audio buffer does not underflow or overflow.

A sample audio post could look like:

```
POST /audiorx.cgi HTTP/1.1
Authorization: Basic <audio:sound base64-encoded>
User-Agent: myclient
Host: mycamera

<ulaw-encoded 8khz mono data>

<end of connection>
```

If eight seconds of audio data needs to be transmitted the host should manage the data to attempt to keep the audio buffer half full, i.e. with two seconds of audio data always waiting to be played. For example:

```
Post two seconds of data from 0 to 2 seconds
Wait one second
Post two seconds of data from 2 to 4 seconds
Wait two seconds
Post two seconds of data from 4 to 6 seconds
Wait two seconds
Post two seconds of data from 6 to 8 seconds
```

12. IQinVision Cameras

12.1. IQ04xx/D4xx/54xx series

All IQeye IQ04xx/D4xx /54xx Series cameras can have the following capabilities.

- Single JPEG image or serverpush Motion-JPEG image stream via HTTP
- Image cropping
- Motion detection with one built-in detection window
- IEEE 802.3af POE power
- PSIA Service Model support
- Aux power input (IQ54 series only)
- Analog output for setting field of view and focusing (IQ54 series only)

The IQeye IQ04xx/D4xx/05xx Series includes the following camera models.

Model	Description
IQ040S	VGA Camera, 30 fps
IQ041S	1.3 MP Camera, 15 fps
IQ042S	2.0 MP Camera, 15 fps
IQD40S	VGA Mini-dome camera, 30 fps
IQD41S	1.3 MP Mini-dome camera, 15 fps
IQD42S	2.0 MP Camera, 15 fps
IQ540S	VGA Camera, 30 fps
IQ541S	1.3 MP Camera, 15 fps
IQ542S	2.0 MP Camera, 15 fps

12.2. IQeye51x series

All IQeye51x series cameras have the following capabilities.

- Single JPEG image or serverpush Motion-JPEG image stream via HTTP
- Simultaneous downsampled image streams
- On-camera motion detection
- IQfocus analog configuration
- IEEE 802.3af POE power
- Aux power input

The IQeye5xx series includes the following camera models.

Model	Description
IQeye501	1.3 MP Camera, 15 fps Note that this camera has been replaced by the IQeye511
IQeye510	Wide Dynamic Range, WVGA Camera, 60 fps
IQeye511	1.3 MP Camera, 15 fps

12.3. IQeye70x/75x/8xx series cameras

All IQeye70x/75x/8xx series cameras have the following capabilities:

- Single JPEG image or serverpush Motion-JPEG image stream via HTTP
- Simultaneous independent image streams
- Image downsampling
- Destination resolution
- Virtual Camera (VCAM) windows
- On-camera motion detection
- Motion bitmap
- IQfocus analog configuration
- IQ7xx cameras have full time analog output
- IEEE 802.3af POE power
- Aux power input

The IQeye70x/75x/8xx series includes the following camera models.

Model	Description
Iqeye701	1.3 MP Camera, 30 fps Note that this camera has been replaced by the IQeye711
Iqeye702	2.0 MP Camera, 20 fps
Iqeye703	3.1 MP Camera, 12 fps
Iqeye705	5.0 MP Camera 8 fps
Iqeye711	1.3MP Camera, 30 fps
Iqeye751	1.3 MP Day/Night Camera, 30 fps
Iqeye752	2.0 MP Day/Night Camera, 20 fps
Iqeye753	3.1 MP Day/Night Camera, 12 fps
Iqeye755	5.0 MP Day/Night Camera, 8 fps
Iqeye802	Sentinel 2.0 MP All-Weather Camera System, 20 fps
Iqeye803	Sentinel 3.1 MP All-Weather Camera System, 12 fps
Iqeye805	Sentinel 5.0 MP All-Weather Camera System, 8 fps
Iqeye811	Sentinel 1.3 MP All-Weather Camera System, 30 fps
IQeye851	Sentinel 1.3 MP All-Weather Camera System, 30 fps
Iqeye852	Sentinel 2.0 MP All-Weather Day/Night Camera System, 20 fps
Iqeye853	Sentinel 3.1 MP All-Weather Day/Night Camera System, 12 fps
Iqeye855	Sentinel 5.0 MP All-Weather Day/Night Camera System, 8 fps

12.4. IQA1xx series cameras

All IQA1xx series Alliance cameras have the following capabilities:

- Single JPEG image or serverpush Motion-JPEG image stream via HTTP
- Simultaneous independent image streams
- Image downsampling
- Destination resolution
- Virtual Camera (VCAM) windows
- On-camera motion detection
- Motion bitmap
- IQfocus analog configuration
- IQA1xx cameras have full time analog output
- IEEE 802.3af POE power
- Aux power input

The IQeye Alliance series includes the following camera models.

Model	Description
IQA10S	VGA Vandal Dome, 30 fps
IQA11S	1.3 MP Vandal Dome, 15 fps
IQA12S	2.0 MP Vandal Dome, 15 fps
IQA13S	3.1 MP Vandal Dome, 12 fps
IQA15S	5.0 MP Vandal Dome, 8 fps

12.5. IQA2xx series cameras

All IQA2xx series Alliance H.264 cameras have the following capabilities:

- Single JPEG image or serverpush Motion-JPEG image stream via HTTP
- H.264 based RTP/RTSP stream at VGA resolution
- Simultaneous independent image streams
- Image downsampling
- Destination resolution
- Virtual Camera (VCAM) windows
- On-camera motion detection
- Motion bitmap
- IQfocus analog configuration
- IQA2xx cameras have full time analog output
- IEEE 802.3af POE power
- Aux power input

The Alliance H.264 enabled series includes the following camera models.

Model	Description
IQA20S	H.264 VGA Vandal Dome, 30 fps
IQA21S	H.264 VGA, MJPEG 1.3 MP Vandal Dome, 15 fps
IQA22S	H.264 VGA, MJPEG 2.0 MP Vandal Dome, 15 fps
IQA23S	H.264 VGA, MJPEG 3.1 MP Vandal Dome, 12 fps
IQA25S	H.264 VGA, MJPEG 5.0 MP Vandal Dome, 8 fp

12.6. IQA3xx series cameras

All IQA3xx series Alliance H.264 cameras have the following capabilities:

- Single JPEG image or serverpush Motion-JPEG image stream via HTTP
- H.264 based RTP/RTSP image data
- “Stream” or “Mode” based H.264 operation
- Stream based H.264 configuration
- On-camera motion detection
- Two-way audio
- PSIA Service Model support
- ONVIF support
- IEEE 802.3af POE power
- Aux power input, 24 VAC or 12-24 VDC

The Alliance IQA3xx series includes the following camera models.

Model	Description
IQA30N	H.264 SD480p, 30 fps
IQA31N	H.264 HD720p, 30 fps
IQA32N	H.264 HD1080p, 30 fps
IQA33N	H.264 3MP - 2048x1536, 15 fps
IQA35N	H.264 5MP – 2560x1920, 10 fps

12.7. IQ73xx/83xx/D3xx/M3xx series cameras

All IQ73xx/83xx/D3xx/M3xx series H.264 enabled cameras have the following capabilities:

- Single JPEG image or serverpush Motion-JPEG image stream via HTTP
- H.264 based RTP/RTSP image data
- "Mode" based H.264 operation
- On-camera motion detection
- PSIA Service Model support
- ONVIF support
- IEEE 802.3af POE power
- Two-way audio support (IQD3xx/IQM3xx models only)
- Analog output for setting field of view and focusing (IQ73xx and IQ83xx models only)
- Aux power input 24 VAC or 12-24 VDC (IQ73xx and IQ83xx models only)

The IQ73xx H.264 enabled series includes the following camera models.

Model	Description
IQ732N	H.264 1080p, 30 fps, day/night
IQ732S	H.264 1080p, 30 fps
IQ832N	H.264 1080p, 30 fps, day/night
IQ832S	H.264 1080p, 30 fps
IQD30S	H.264 480p, 30 fps
IQD31S	H.264 720p, 30 fps
IQD32S	H.264 1080p, 15 fps
IQM30S	H.264 480p, 30 fps
IQM31S	H.264 720p, 30 fps
IQM32S	H.264 1080p, 15 fps

13. Host Based Interfacing

13.1. IQstream sample client software

IQinVision supplies IQstream, a Visual Studio application that demonstrates how to request, parse and display a serverpush MJPEG stream from IQeye cameras. Please download the application and/or Visual Studio project from the "Support->FAQs & Downloads" page on the IQinVision web site at http://www.iqeye.com/faq.html.

13.2. ActiveX control

IQinVision supplies an ActiveX component that can be used to interface to cameras in a Microsoft Visual Studio or web page environment. This component has been optimized to transport and display a JPEG serverpush stream very efficiently. In addition, the control provides a mechanism for reading and writing OID values. This allows a host based application to get and set configuration values on the camera.

Please download the ActiveX control from the "Support->FAQs & Downloads" page on the IQinVision web site at http://www.iqeye.com/faq.html. The Readme file included in the zip archive contains the most current documentation for the control.

13.3. Win32 .Net interface

IQinVision supplies NetStream, a simple streaming client implemented using C# in the .Net 2.0 environment. Please download this sample program from the Support->FAQs & Downloads" page on the IQinVision web site at http://www.iqeye.com/faq.html.

14. ONVIF Support

The ONVIF open industry forum developed a global standard for the interface of IP-based physical security products.

14.1. Supported ONVIF commands

The following table shows commands supported by selected IQinVision cameras.

Number	ONVIF Command List	Supported
1	GetProfile	Yes
2	GetProfiles	Yes
3	GetVideoEncoderConfigurations	Yes
4	SetVideoEncoderConfigurations	Yes
5	GetVideoEncoderConfigurationOptions	No
6	GetStreamUri	Yes
7	GetDeviceInformation	Yes
8	SetSystemDateAndTime	Yes
9	GetSystemDateAndTime	Yes
10	SetSystemFactoryDefault	Yes
11	SystemReboot	Yes
12	GetScopes	Yes
13	AddScopes	Yes
14	RemoveScopes	Yes
15	GetWsdlUrl	Yes
16	GetCapabilities	Yes
17	GetHostname	Yes
18	SetHostname	Yes
19	GetDNS	Yes
20	SetDNS	Yes
21	GetNTP	Yes
22	SetNTP	Yes
23	GetNetworkInterfaces	Yes
24	GetNetworkProtocols	Yes
25	SetNetworkProtocols	No
26	GetImagingSettings	No
27	SetImagingSettings	No
28	GetOptions	?
29	SetNetworkProtocols	Yes (duplicate entry)
30	GetNetworkProtocols	Yes (duplicate entry)
31	GetVideoSource	No
32	SetVideoSource	No
33	GetVideoSourceConfigurations	No
34	SetVideoSourceConfigurations	No
35	frame rate setting	No
36	VMD trigger	No
37	alarm in / out	No

15. PSIA Support

The Physical Security Interoperability Alliance (PSIA) has published a PSIA Service Model. This model describes the methods used for service discovery, device configuration and streaming media support.

15.1. Service requirements

The following table shows the functional groups supported by PSIA compliant IQinVision cameras.

The check boxes denote "support required" PSIA commands. The green colored fields show the PSIA commands supported by selected IQinVision cameras.

REQ	Service URL	Notes
✓	/PSIA	
✓	/PSIA/System	
	/PSIA/System/Storage	Not all IP media devices support storage.
✓	/PSIA/System/Network	
	/PSIA/System/IO	
	/PSIA/System/Audio	
	/PSIA/System/Video	
	/PSIA/System/Serial	
	/PSIA/Diagnostics	
✓	/PSIA/Security	
	/PSIA/Security/AAA	
	/PSIA /Streaming	
	/PSIA /PTZ	
	/PSIA /Custom/MotionDetection	
	/PSIA /Custom/Event	

Note: Not all IQinVision cameras support the PSIA Service Model. Please see each camera description to determine supported status.

15.2. Resource requirements

The following resources are required for each implemented service. The check boxes denote "support required" PSIA commands. The green colored fields show the PSIA commands supported by IQinVision cameras.

15.2.1. /PSIA root service

REQ	Command	GET	PUT	POST	DEL
✓	index	✓			
✓	indexr	✓			
✓	description	✓			

15.2.2. /PSIA/System

REQ	Command	GET	PUT	POST	DEL
✓	reboot		✓		
✓	updateFirmware		✓		
✓	configurationData	✓	✓		
✓	factoryReset		✓		
✓	deviceInfo	✓	✓		
	supportReport	✓			
✓	status	✓			
	time	✓	✓		
	time/localTime	✓	✓		
	time/timeZone	✓	✓		
	time/ntpServers	✓	✓	✓	✓
	time/ntpServers/<ID>	✓	✓		✓
	logging	✓	✓		
	logging/messages	✓			

15.2.3. /PSIA/System/Storage

REQ	Command	GET	PUT	POST	DEL
	volumes	✓			
	volumes/<ID>	✓			
	volumes/<ID>/status	✓			
	volumes/<ID>/format		✓		
	volumes/<ID>/files	✓			✓
	volumes/<ID>/files/<ID>	✓			✓
	volumes/<ID>/files/<ID>/data	✓			

15.2.4. /PSIA/System/Network

REQ	Command	GET	PUT	POST	DEL
✓	interfaces	✓			
✓	interfaces/<ID>	✓	✓		
✓	interfaces/<ID>/ipAddress	✓	✓		
	interfaces/<ID>/wireless	✓	✓		
	interfaces/<ID>/ieee802.1x	✓	✓		
	interfaces/<ID>/ipFilter	✓	✓		
	interfaces/<ID>/ipFilter/filterAddresses	✓	✓	✓	✓
	interfaces/<ID>/ipFilter/filterAddresses/<ID>	✓	✓		✓
	interfaces/<ID>/snmp	✓	✓		
	interfaces/<ID>/snmp/v2c	✓	✓		
	interfaces/<ID>/snmp/v2c/trapReceivers	✓	✓	✓	✓
	interfaces/<ID>/snmp/v2c/trapReceivers/<ID>	✓	✓		✓
	interfaces/<ID>/snmp/advanced	✓	✓		
	interfaces/<ID>/snmp/advanced/users	✓	✓	✓	✓
	interfaces/<ID>/snmp/advanced/users/<ID>	✓	✓		✓
	interfaces/<ID>/snmp/advanced/notificationFilters	✓	✓	✓	✓
	interfaces/<ID>/snmp/advanced/notificationFilters/<ID>	✓	✓		✓
	interfaces/<ID>/snmp/advanced/notificationReceivers	✓	✓	✓	✓
	interfaces/<ID>/snmp/advanced/notificationReceivers/<ID>	✓	✓		✓
	interfaces/<ID>/snmp/v3	✓	✓		
	interfaces/<ID>/qos	✓	✓		
	interfaces/<ID>/qos/cos	✓	✓	✓	✓
	interfaces/<ID>/qos/cos/<ID>	✓	✓		✓
	interfaces/<ID>/qos/dscp	✓	✓	✓	✓
	interfaces/<ID>/qos/dscp/<ID>	✓	✓		✓
✓	interfaces/<ID>/discovery	✓	✓		
	interfaces/<ID>/syslog	✓	✓		
	interfaces/<ID>/syslog/servers	✓	✓	✓	✓
	interfaces/<ID>/syslog/servers/<ID>	✓	✓		✓

15.2.5. /PSIA/System/IO

REQ	Command	GET	PUT	POST	DEL
	status	✓			
	inputs	✓			
	inputs/<ID>	✓	✓		
	inputs/<ID>/status	✓			
	outputs	✓			
	outputs/<ID>	✓	✓		
	outputs/<ID>/trigger		✓		
	outputs/<ID>/status	✓			

15.2.6. /PSIA/System/Audio

REQ	Command	GET	PUT	POST	DEL
	channels	✓			
	channels/<ID>	✓	✓		

15.2.7. /PSIA/System/Video

REQ	Command	GET	PUT	POST	DEL
	overlayImages	✓		✓	✓
	overlayImages/<ID>	✓	✓		✓
✓	inputs	✓			
✓	inputs/channels	✓			
✓	inputs/channels/<ID>	✓	✓		
	inputs/channels/<ID>/focus		✓		
	inputs/channels/<ID>/iris		✓		
	inputs/channels/<ID>/lens	✓			
	inputs/channels/<ID>/overlays	✓	✓		✓
	inputs/channels/<ID>/overlays/text	✓	✓	✓	✓
	inputs/channels/<ID>/overlays/text/<ID>	✓	✓		✓
	inputs/channels/<ID>/overlays/image	✓	✓	✓	✓
	inputs/channels/<ID>/overlays/image/<ID>	✓	✓		✓
	inputs/channels/<ID>/privacyMask	✓	✓		
	inputs/channels/<ID>/privacyMask/regions	✓	✓	✓	✓
	inputs/channels/<ID>/privacyMask/regions/<ID>	✓	✓		✓

15.2.8. /PSIA/System/Serial

REQ	Command	GET	PUT	POST	DEL
	ports	✓			
	ports/<ID>	✓	✓		
	ports/<ID>/command		✓		

15.2.9. /PSIA/Diagnostics

REQ	Command	GET	PUT	POST	DEL
	commands	✓		✓	✓
	commands/<ID>	✓			✓

15.2.10. /PSIA/Security

REQ	Command	GET	PUT	POST	DEL
	srtpMasterKey	✓	✓		
	deviceCertificate	✓	✓		

15.2.11. /PSIA/Security/AAA

REQ	Command	GET	PUT	POST	DEL
✓	users	✓	✓	✓	✓
✓	users/<ID>	✓	✓		✓
	certificate	✓	✓		
	adminAccesses	✓	✓	✓	✓
	adminAccesses/<ID>	✓	✓		✓

15.2.12. /PSIA/Streaming

REQ	Command	CAP	GET	PUT	POST	DEL
✓	status		✓			
✓	channels		✓	✓	✓?	✓?
✓	channels/<ID>		✓	✓		✓?
✓	channels/<ID>/status		✓			
	channels/<ID>/http		✓			
	channels/<ID>/picture		✓			
	channels/<ID>/requestKeyFrame			✓		

15.2.13. /PSIA/PTZ

REQ	Command	GET	PUT	POST	DEL
	channels	✓	✓	✓?	✓?
	channels/<ID>	✓	✓		✓?
	channels/<ID>/homePosition		✓		
	channels/<ID>/continuous		✓		
	channels/<ID>/momentary		✓		
	channels/<ID>/relative		✓		
	channels/<ID>/absolute		✓		
	channels/<ID>/digital		✓		
	channels/<ID>/status	✓			
	channels/<ID>/presets	✓	✓	✓	✓
	channels/<ID>/presets/<ID>	✓	✓		✓
	channels/<ID>/presets/<ID>/goto		✓		
	channels/<ID>/patrols	✓	✓	✓	✓
	channels/<ID>/patrols/status	✓			
	channels/<ID>/patrols/<ID>	✓	✓		✓
	channels/<ID>/patrols/<ID>/start		✓		
	channels/<ID>/patrols/<ID>/stop		✓		
	channels/<ID>/patrols/<ID>/pause		✓		
	channels/<ID>/patrols/<ID>/status	✓			
	channels/<ID>/patrols/<ID>/schedule	✓	✓		

15.2.14. /PSIA/Custom/MotionDetection

REQ	Command	GET	PUT	POST	DEL
		✓			
	<ID>	✓	✓		
	<ID>/regions	✓	✓	✓	✓
	<ID>/regions/<ID>	✓	✓		✓

15.2.15. /PSIA/Custom/Event

REQ	Command	GET	PUT	POST	DEL
	trigger	✓	✓		
	trigger/triggers	✓	✓	✓	✓
	trigger/triggers/<ID>	✓	✓		✓
	trigger/triggers/<ID>/notifications	✓	✓	✓	✓
	trigger/triggers/<ID>/notifications/<ID>	✓	✓		✓
	trigger/schedule	✓	✓		
	notification	✓	✓		
	notification/mailing	✓	✓	✓	✓
	notification/mailing/<ID>	✓	✓		✓
	notification/ftp	✓	✓	✓	✓
	notification/ftp/<ID>	✓	✓		✓
	notification/httpHost	✓	✓	✓	✓
	notification/httpHost/<ID>	✓	✓		✓
	notification/alertStream	✓			

A. Technical Support

If you are experiencing problems with your IQeye camera, please visit the technical support section of the IQinVision web site, located at http://www.iqeye.com

If you cannot find an answer to your problem by referencing either the *Troubleshooting* section of the IQeye Reference Manual or the support FAQs on the IQinVision web site, please contact an IQinVision representative for technical support.

IQinVision
email: support@iqeye.com
Phone: 949-369-8100
toll free: 877-850-0805

When you contact IQinVision Technical Support, please make sure that you have the following information available:

- Your name, phone number, and email address,
- Your company name, if applicable, and its address and telephone number,
- Your camera's hardware number (printed on the label affixed to the rear cap of the camera). The hardware number is also known as a MAC address or Ethernet address.
- Your camera's firmware revision which is visible on the security settings web page
- A detailed description of the problem.