AVT GigE

Camera and Driver Attributes

Firmware 1.50

June 21, 2012

Allied Vision Technologies Canada Inc.
101-3750 North Fraser Way
Burnaby, BC
V5J 5E9 / Canada

Notes

1) This is the master document for all camera models. Not all attributes are available on all cameras or firmware versions. For 3rd party users, see the camera XML file. For PvAPI users, see the *PvAttrIsAvailable* function call.

2) For PvAPI users, attribute type is given: Enum, Float32, Uint32, String, or Command. See the corresponding *PvAttrEnum___*, *PvAttrFloat32___*, *PvAttrUint32___*, *PvAttrString___*, *PvCommandRun* calls.

3) Uint32 and Float32 ranges: where camera dependent, see camera user manual, or see slider control in SampleViewer. PvAPI users see *PvAttrRangeEnum*, *PvAttrRangeFloat32* calls.

4) R/W = attribute is read/write. R/C = attribute is read only and constant. R/V = attribute is read only and volatile, can change at any time.

Camera Attributes

Camera Attributes affect settings on the camera itself – vs. Driver Attributes which affect the host PvAPI driver.

Acquisition

Trigger

AcqEnd

AcqEndTriggerEvent – Enum – R/W

If AcqEndTriggerMode = SyncIn1/2, determines which SyncIn1/2 electrical signal initiates trigger.

EdgeRising	rising edge trigger
EdgeFalling	falling edge trigger
EdgeAny	rising or falling edge
LevelHigh	active high signal
LevelLow	active low signal

AcqEndTriggerMode – Enum – R/W

Is the end of acquisition initiated by an external hardware trigger.

SyncIn1	trigger at SyncIn1 to be associated with this control
SyncIn2	trigger at SyncIn2 to be associated with this control
Disabled	no external trigger. Acquisition must be stopped with the AcquisitionStop API command.

AcqRec

An AcqStart hardware trigger signal, or the AcquisitionStart command, must be received before your AcqRec trigger. See AcquisitionMode = Recorder.

AcqRecTriggerEvent – Enum – R/W

If AcqRecTriggerMode = SyncIn1/2, determines which SyncIn1/2 electrical signal initiates trigger.

EdgeRising	rising edge trigger
EdgeFalling	falling edge trigger
EdgeAny	rising or falling edge
LevelHigh	active high signal
LevelLow	active low signal

AcqRecTriggerMode – Enum – R/W

Is the recorder mode trigger event initiated by an external hardware trigger.

SyncIn1	trigger at SyncIn1 to be associated with this control
SyncIn2	trigger at SyncIn2 to be associated with this control
Disabled	no external trigger. Unlike AcqStart and AcqEnd, there is no API command trigger option for a recording event.

AcqStart

Controls relating to the start of an acquisition stream. Frames are triggered within this acquisition stream (see FrameStart).

AcqStartTriggerEvent – Enum – R/W

If AcqStartTriggerMode = SyncIn1/2, determines which SyncIn1/2 electrical signal initiates trigger.

EdgeRising	rising edge trigger
EdgeFalling	falling edge trigger
EdgeAny	rising or falling edge
LevelHigh	active high signal
LevelLow	active low signal

AcqStartTriggerMode – Enum – R/W

Is the start of acquisition initiated by an external hardware trigger.

SyncIn1	trigger at SyncIn1 to be associated with this control
SyncIn2	trigger at SyncIn2 to be associated with this control
Disabled	no external trigger. Acquisition must be started with the AcquisitionStart API command.

FrameRate – Float32 – R/W

Range: [cam dependent]. Units: Hz. When FrameStartTriggerMode is set to FixedRate, this control specifies the frame rate.

FrameStart

Controls relating to the triggering of frames within an acquisition stream.

FrameStartTriggerDelay – Uint32 – R/W

Range: [0 - cam dependent]. Units: µs. Start-of-image is delayed FrameStartTriggerDelay microseconds after receiving an external trigger event. This feature is only valid when FrameStartTriggerMode is set to external trigger (i.e. SyncIn1, SyncIn2). Useful when using a common trigger to synch with a strobe lighting source, which will have some fixed setup time.

FrameStartTriggerEvent – Enum – R/W

If FrameStartTriggerMode = SyncIn1/2, determines which SyncIn1/2 electrical signal initiates trigger.

EdgeRising	rising edge trigger
EdgeFalling	falling edge trigger
EdgeAny	rising or falling edge
LevelHigh	active high signal
LevelLow	active low signal

FrameStartTriggerMode – Enum – R/W

Determines how an image is initiated once AcquisitionStart called, or AcqStart hardware trigger set. For *Freerun* and *FixedRate* the first frame is synchronized to AcquisitionStart/AcqStart trigger.

Freerun	frame triggers generated on-camera, at maximum supported frame rate depending on the exposure time and region of interest size.
SyncIn1	external trigger SyncIn1
SyncIn2	external trigger SyncIn2
FixedRate	frame triggers generated on-camera, at frame rate defined by FrameRate attribute.
Software	software initiated frame trigger. See FrameStartTriggerSoftware command.

FrameStartTriggerOverlap – Enum – R/W

Possible values: *Off, PreviousFrame.* When *Off*, any external trigger received before *FrameTriggerReady* signal is high is ignored. When *PreviousFrame*, any external trigger received before *FrameTriggerReady* is latched and used to trigger the next frame. Default is *Off.* Does not work with Software triggering. Only external.

FrameStartTriggerSoftware – Command

Triggers an image. Valid when FrameStartTriggerMode = Software.

AcquisitionAbort – Command

Software command to stop camera from receiving frame triggers, plus aborts any currently exposing image.

AcquisitionFrameCount – Uint32 – R/W

Range: [1 – 65535]. Units: Frames. The number of frames to capture in a limited sequence of images. Used with AcquisitionMode = MultiFrame and Recorder. In Recorder mode, AcquisitionFrameCount cannot exceed StreamHoldCapacity.

AcquisitionMode – Enum – R/W

Determine how many frame triggers the camera receives after acquisition start event.

Continuous	The camera will continuously receive frame triggers.

SingleFrame	The camera will only receive a single frame trigger event. Further frame triggers will be ignored until acquisition is stopped and restarted.
MultiFrame	The camera will receive AcquisitionFrameCount number of frame triggers. Further frame triggers will be ignored until acquisition is stopped and restarted.
Recorder	The camera will continuously capture images into camera memory, but will not send them to the host until an AcqRec trigger signal is received. Further AcqRec trigger events will be ignored until acquisition is stopped and restarted. This feature allows returning RecorderPreEventCount number of frames **before** the trigger event, and AcquisitionFrameCount minus RecorderPreEventCount frames after the trigger. When AcqRec trigger is received, the currently imaging/aquiring image will complete as normal, and then at least one more image will be taken. Camera memory is a circular buffer, once it is full, it starts overwriting images.

AcquisitionStart – Command

Software command to start camera receiving frame triggers. Valid when AcqStartTriggerMode = disabled. See FrameStartTriggerMode.

AcquisitionStop – Command

Software command to stop camera from receiving frame triggers. Valid when AcqEndTriggerMode = disabled. See FrameStartTriggerMode.

RecorderPreEventCount – Uint32 – R/W

Range: [0* – 65535]. Units: Frames. The number of images returned before the AcqRec trigger event, with AquisitionFrameCount minus RecorderPreEventCount images being returned after the trigger event. Valid only when AcquisitionMode equals Recorder.

*Note: at least one image must be captured after the AcqRec trigger event. That is, you cannot set RecorderPreEventCount = 1, AcquisitionFrameCount = 1.

ConfigFile

AVT's GigE cameras are capable of storing a number of user-specified configurations within the camera's non-volatile memory. These saved configurations can be used to define the power-up settings of the camera or to quickly switch between a number of predefined settings.

NOTE: Only **camera** attributes (those attributes listed in this document under the Camera Attributes section) can be saved to a camera ConfigFile. **Driver** attributes cannot be saved.

ConfigFileIndex – Enum – R/W

Possible values: *Factory, 1, 2, 3, 4, 5*. The index number corresponding to the configuration set that you are currently working with.

ConfigFileLoad – Command

Loads settings saved in camera non-volatile memory indicated by ConfigFileIndex to the current camera settings.

ConfigFilePowerup – Enum – R/W

Possible values: *Factory, 1, 2, 3, 4, 5*. The saved configuration loaded when the camera powers up.

ConfigFileSave – Command

Saves the current camera settings to camera non-volatile memory indicated by ConfigFileIndex. The Factory setting cannot be overwritten.

Controls

ColorTransformationControl

Applied post-WhiteBalance, post-bayer interpolation. Only available on color PixelFormats noted with on-camera interpolation. Each pixel has a R, G, B channel value. Color transformation takes these input channel values and multiplies them by a 3x3 matrix, allowing for highly configurable color.

$$\begin{pmatrix} R_{out} \\ G_{out} \\ B_{out} \end{pmatrix} = \begin{pmatrix} CTV_{RR} & CTV_{RG} & CTV_{RB} \\ CTV_{GR} & CTV_{GG} & CTV_{GB} \\ CTV_{BR} & CTV_{BG} & CTV_{BB} \end{pmatrix} \times \begin{pmatrix} R_{in} \\ G_{in} \\ B_{in} \end{pmatrix}$$

Figure 1: Color transformation equation.
See ColorTransformationValue## attributes.

ColorTransformationMode – Enum – R/W

Off	No color transformation is done.
Manual	Manually set ColorTransformationValue matrix coefficients.
Temp6500K	Colors optimized for a surrounding color temperature 6500 K.

ColorTransformationValueBB – Float32 – R/W

Range: [0.000 – 2.000]. Blue multiplicative factor applied to blue input channel.

ColorTransformationValueBG – Float32 – R/W

Range: [0.000 – 2.000]. Green multiplicative factor applied to blue input channel.

ColorTransformationValueBR – Float32 – R/W

Range: [0.000 – 2.000]. Red multiplicative factor applied to blue input channel.

ColorTransformationValueGB – Float32 – R/W

Range: [0.000 – 2.000]. Blue multiplicative factor applied to green input channel.

ColorTransformationValueGG – Float32 – R/W

Range: [0.000 – 2.000]. Green multiplicative factor applied to green input channel.

ColorTransformationValueGR – Float32 – R/W

Range: [0.000 – 2.000]. Red multiplicative factor applied to green input channel.

ColorTransformationValueRB – Float32 – R/W

Range: [0.000 – 2.000]. Blue multiplicative factor applied to red input channel.

ColorTransformationValueRG – Float32 – R/W

Range: [0.000 – 2.000]. Green multiplicative factor applied to red input channel.

ColorTransformationValueRR – Float32 – R/W

Range: [0.000 – 2.000]. Red multiplicative factor applied to red input channel.

DSP

The automatic exposure, gain, WhiteBalance, and iris features can be configured to respond only to a subregion within the image scene. This feature can be used to choose a subregion that will 'meter' the rest of the image. This feature works like the region metering on a photographic camera.

DSPSubregionBottom – Uint32 – R/W

Range: [0 – 4294967295*]. Units: Rows from top edge of full image. Defines the bottom edge of the DSP subregion. *Note: Any value beyond sensor height (see camera user manual) limited to sensor height.

DSPSubregionLeft – Uint32 – R/W

Range: [0 – 4294967295*]. Units: Columns from left edge of full image. Defines the left edge of the DSP subregion. *Note: Any value beyond sensor width (see camera user manual) limited to sensor width.

DSPSubregionRight – Uint32 – R/W

Range: [0 – 4294967295*]. Units: Columns from left edge of full image. Defines the right edge of the DSP subregion. *Note: Any value beyond sensor width (see camera user manual) limited to sensor width.

DSPSubregionTop – Uint32 – R/W

Range: [0 – 4294967295*]. Units: Rows from top edge of full image. Defines the top edge of the DSP subregion. *Note: Any value beyond sensor height (see camera user manual) limited to sensor height.

DefectMask

This feature is only available on the GE4000 and GE4900 cameras. The standard model of these cameras uses Class 2 sensors which can have column defects. Class 1 and Class 0 sensors are available for these cameras which do not require any column masking.

DefectMaskColumnEnable – Enum – R/W

Possible values: *On, Off.* Defect masking replaces defective columns with interpolated values based on neighboring columns.

EdgeFilter – Enum – R/W

Image sharpness/blur. Applied post-bayer interpolation. Only available on color PixelFormats noted with on-camera interpolation.

Smooth2	Most blur.
Smooth1	Slight blur.
Off	No blur or sharpness applied.
Sharpen1	Slight sharp.
Sharpen2	Most sharp.

Exposure

Auto

NOTE: The camera must be acquiring images in AcquisitionMode = Continuous in order for the auto algorithm to update.

Auto algorithms use information from the camera's current image and apply the following settings to the next image. Large changes in scene lighting may require several frames for the algorithm to stabilize.

If using ExposureMode = Auto, and GainMode = Auto simultaneously, priority is given to changes in exposure until ExposureAutoMax is reached, at which point priority is given to changes in gain. Adding simultaneous IrisMode = Video/DCIris/PIrisAuto results in undefined, "race to target" behavior.

ExposureAutoAdjustTol – Uint32 – R/W

Range: [0 – 50]. Units: percent. Tolerance in variation from ExposureAutoTarget in which the auto exposure algorithm will not respond. Can be used to limit exposure setting changes to only larger variations in scene lighting.

ExposureAutoAlg – Enum – R/W

The following algorithms can be used to calculate auto-exposure:

Mean The arithmetic mean of the histogram of the current image is compared to ExposureAutoTarget, and the next image adjusted in exposure time to meet this target. Bright areas are allowed to saturate.

FitRange The histogram of the current image is measured, and the exposure time of the next image is adjusted so bright areas are not saturated. Generally, the Mean setting is preferred.

ExposureAutoMax – Uint32 – R/W

Range: [cam dependent – 60000000]. Units: µs. The upper bound to the exposure setting in autoexposure mode. This is useful in situations where frame rate is important. This value would normally be set to something less than 1x10^6/(desired frame rate).

ExposureAutoMin – Uint32 – R/W

Range: [cam dependent – 60000000]. Units: µs. The lower bound to the exposure setting in autoexposure mode.

ExposureAutoOutliers – Uint32 – R/W

Range: [0 – 1000]. Units: 0.01%, i.e. 1000 = 10%. The total pixels from top of the distribution that are ignored by the auto exposure algorithm.

ExposureAutoRate – Uint32 – R/W

Range: [1 – 100]. Units: percent. The rate at which the auto exposure function changes the exposure setting.

ExposureAutoTarget – Uint32 – R/W

Range: [0 – 100]. Units: percent. The general lightness or darkness of the auto exposure feature; specifically the target mean histogram level of the image, 0 being black, 100 being white.

ExposureMode – Enum – R/W

Manual The camera exposure time is fixed by ExposureValue parameter.

Auto The exposure time will vary continuously according to the scene illumination. The Auto exposure function operates according to the Auto and DSP controls

AutoOnce A command. The exposure will be set once according to the scene illumination and then remain at that setting even when the scene illumination changes. The AutoOnce function operates according to the Auto and DSP controls.

External When ExposureMode is set to External the exposure time will be controlled by an external signal appearing on SyncIn1 or SyncIn2. In order for this feature to work, the parameter FrameStartTriggerMode must be set to SyncIn1 or SyncIn2.

ExposureValue – Uint32 – R/W

Range: [cam dependent – 60000000]. Units: microseconds. The sensor integration time.

Gain

Auto

NOTE: The camera must be acquiring images in AcquisitionMode = Continuous in order for the auto algorithm to update.

Auto algorithms use information from the camera's current image and apply the following settings to the next image. Large changes in scene lighting may require 2-3 frames for the algorithm to stabilize.

If using ExposureMode = Auto, and GainMode = Auto simultaneously, priority is given to changes in exposure until ExposureAutoMax is reached, at which point priority is given to changes in gain. Adding simultaneous Video/DCIris/PIrisAuto results in undefined, "race to target" behavior.

GainAutoAdjustTol – Uint32 – R/W

Range: [0 – 50]. Units: percent. Sets a tolerance in variation from GainAutoTarget in which the auto exposure algorithm will not respond. Can be used to limit gain setting changes to only larger variations in scene lighting.

GainAutoMax – Uint32 – R/W

Range: [0 – cam dependent]. Units: dB. Sets the upper bound to the gain setting in Auto gain mode.

GainAutoMin – Uint32 – R/W

Range: [0 – cam dependent]. Units: dB. Sets the lower bound to the gain setting in Auto gain mode. Normally this number would be set to zero.

GainAutoOutliers – Uint32 – R/W

Range: [0 – 1000]. Units: 0.01%, i.e. 1000 = 10%. The total pixels from top of the distribution that are ignored by the auto gain algorithm.

GainAutoRate – Uint32 – R/W

Range: [1 – 100]. Units: percent. The rate at which the auto gain function changes the gain setting.

GainAutoTarget – Uint32 – R/W

Range: [0 – 100]. Units: percent. The general lightness or darkness of the Auto gain feature. A percentage of the maximum GainValue.

GainMode – Enum – R/W

Manual	The camera gain is fixed by GainValue parameter.
Auto	The gain will vary continuously according to the scene illumination. The Auto function operates according to the Auto and DSP controls.
AutoOnce	A command. The gain will be set once according to the scene illumination and then remain at that setting even when the scene illumination changes. The AutoOnce function operates according to the Auto and DSP controls.

GainValue – Uint32 – R/W

Range: [0 – cam dependent]. Units: dB. $G_{dB} = 20 \log_{10}(V_{out}/V_{in})$. The gain setting applied to the sensor. Default gain is zero, and gives the best image quality. However, in low light situations, it may be necessary to increase the gain setting.

Gamma – Float32 – R/W

Range: [0.250 - 3.000]. Units: Output = $(\text{Input})^{\text{Gamma}}$. Nonlinear brightness control.

Hue – Float32 – R/W

Range: [-40.000 – 40.000]. Units: Degrees. Alters color of image without altering white balance. Takes float input, although rounds to integer. Applied post-bayer interpolation. Only available on color PixelFormats noted with on-camera interpolation.

LensDrive

Open loop DC 3 axis lens control.

LensDriveCommand – Enum – R/W

Setting to any non-Stop value will execute the function for LensDriveDuration and and then return to Stop.

Stop	No action.
IrisTimedOpen	Open lens iris.
IrisTimedClose	Close lens iris.
FocusTimedNear	Shorten working distance.
FocusTimedFar	Lengthen working distance.
ZoomTimedIn	Zoom in.
ZoomTimedOut	Zoom out.

LensDriveDuration – Uint32 – R/W

Range: [0 – 5000]. Units: µs. Duration of LensDriveCommand to lens.

LensVoltage – Uint32 – R/V

Range: [0 – 12000]. Units: mV. Reports the lens power supply voltage.

LensVoltageControl – Uint32 – R/W

Range: [0 – 1200012000]. Units: mV * 100001. E.g. 8 V = 800008000. Lens power supply voltage control. If a bad value is written this control resets to 0. This is done to prevent users inadvertently setting an inappropriate voltage, possibly damaging the lens. See lens documentation for appropriate voltage level.

Iris

Auto iris lens support. Supported auto-iris lens types (camera dependant): video, DC, P-iris. GT series detects lens type on power up. DC settings will not apply if P-Iris lens connected. P-Iris settings will not apply if DC iris lens connected.

The camera must be acquiring images in AcquisitionMode = Continuous in order for the auto algorithm to update. The auto iris algorithm calculates IrisAutoTarget based on information of the current image, and applies this to the next image. Large changes in scene lighting may require 2-3 frames for the algorithm to stabilize.

Adding simultaneous GainMode = Auto, or ExposureMode = Auto, to IrisMode = Video/DCIris/PIrisAuto results in undefined, "race to target" behavior.

IrisAutoTarget – Uint32 – R/W

Range [0 – 100]. Units: percent. Controls the general lightness or darkness of the auto iris feature; specifically the target mean histogram level of the image, 0 being black, 100 being white.

IrisMode – Enum – R/W

Sets the auto-iris mode.

Disabled	Disable auto-iris.
Video	Enable video auto-iris. Video-type lenses only.
VideoOpen	Fully open the iris. Video-type lenses only.
VideoClosed	Full close the iris. Video-type lenses only.
PIrisAuto	Enable P auto-iris. P-Iris lenses only.

PIrisManual	Manually control iris via LensPIrisPosition attribute. P-Iris lenses only.
DCIris	Enable DC auto-iris. DC-Iris lenses only.

IrisVideoLevel – Uint32 – R/V

Dependant on lens type:

- Video-type lenses: Range: [0 – 150]. Units: 10 mV units. Video-type lenses have a reference voltage. When a voltage larger than this reference voltage is applied to the lens, the iris closes. When a voltage is applied less than this reference voltage, the iris opens.
- P-iris lenses: Range: [0-100]. Attempts to match IrisAutoTarget.
- DC-iris lenses: Range: [0-100]. Attempts to match IrisAutoTarget.

IrisVideoLevelMax – Uint32 – R/W

Range: [0 – 150]. Units: 10 mV units. Video-type lenses only. Limits the maximum driving voltage for closing the lens iris. Typically this will be 150.

IrisVideoLevelMin – Uint32 – R/W

Range: [0 – 150]. Units: 10 mV units. Video-type lenses only. Limits the minimum driving voltage for opening the lens iris. Typically this will be 0.

LensDCIris

DC Iris lenses only.

LensDCDriveStrength – Uint32 – R/W

Range: [0 – 50]. Lens drive voltage. Altering this changes the speed at which a DC-Iris lens operates. The lower the value, the slower the lens operates. Too high a value may result in iris oscillation. The recommended value is lens dependant. Larger lenses typically require a larger drive voltage.

LensPIris

P-Iris lenses only. P-Iris allows discrete iris positions using an internal lens stepping motor. See GT camera user manual for a list of PIris supported lenses, along with their LensPIrisFrequency and LensPIrisNumSteps specifications.

LensPIrisFrequency – Uint32 – R/W

Stepping motor drive rate. Lens dependent. Use value defined in GT camera user manual, or contact lens manufacturer.

LensPIrisNumSteps – Uint32 – R/W

Maximum number of discrete iris/aperture positions. Use value defined in GT camera user manual, or contact lens manufacturer.

LensPIrisPosition – Uint32 – R/W

Range: [0 – LensPIrisNumSteps] Iris/aperture position. Manually control iris in PIrisManual mode, or read back iris position in PIrisAuto mode. 0 = fully open, LensPIrisNumSteps = fully closed. Values greater than LensPIrisNumSteps are ignored/not written.

Saturation – Float32 – R/W

Range: [0.000 – 2.000]. Alters color intensity. 0.000 = monochrome. 1.000 = (default) no saturation or de-saturation. 2.000 = full saturation. Applied post-bayer interpolation. Only available on color PixelFormats noted with on-camera interpolation.

SubstrateVoltage

VsubValue – Uint32 – R/C

Units: mV. Factory use only. CCD substrate voltage. Optimized at factory for each sensor.

WhiteBalance

Unlike Hue or ColorTranformationControl, this is a pre-bayer interpolation gain adjustment. Applies to all color PixelFormats.

Auto

Auto algorithms use information from the camera's current image and apply the following settings to the next image. I.e. the camera must be acquiring images in order for the auto algorithm to update. Large changes in scene lighting may require 2-3 frames for the algorithm to stabilize.

WhitebalAutoAdjustTol – Uint32 – R/W

Range: [0 – 50]. Units: percent. A threshold. Sets a range of averaged scene color changes in which the automatic white balance will not respond. Used to limit white balance setting changes to only larger variations in average scene color.

WhitebalAutoRate – Uint32 – R/W

Range: [1 – 100]. Units: percent. Determines how fast the auto white balance algorithm updates.

WhitebalMode – Enum – R/W

Manual Auto white balance is off. White balance can be adjusted directly by changing the WhitebalValueRed and WhitebalValueBlue parameters.

Auto White balance will continuously adjust according to the current scene. The Auto function operates according to the Auto and DSP controls

AutoOnce A command. The white balance will be set once according to the scene illumination and then remain at that setting even when the scene illumination changes. The AutoOnce function operates according to the Auto and DSP controls

WhitebalValueRed – Uint32 – R/W

Range: [cam dependent]. Units: percent. Gain applied to all red pixels on the CCD, pre-interpolation. 100% = no gain applied. Each camera model calibrated with a different factory default.

Note: there is no WhitebalValueGreen, as this is the luminance/reference channel. To increase/decrease green, decrease/increase red+blue accordingly.

WhitebalValueBlue – Uint32 – R/W

Range: [cam dependent]. Units: percent. Gain applied to all blue pixels on the CCD, pre-interpolation. 100% = no gain applied. Each camera model calibrated with a different factory default.

Note: there is no WhitebalValueGreen, as this is the luminance/reference channel. To increase/decrease green, decrease/increase red+blue accordingly.

DeviceStatus

DeviceTemperatureMainboard – Float32 – R/V

Units: Degrees Celsius. Camera internal temperature measured at the internal control board.

EventControls

EventID

All the events supported by the camera:

EventAcquisitionStart – Uint32 – R/C	40000
EventAcquisitionEnd – Uint32 –R/C	40001
EventFrameTrigger – Uint32 – R/C	40002
EventExposureEnd – Uint32 – R/C	40003
EventAcquisitionRecordTrigger – Uint32 – R/C	40004
EventPtpSyncLost – Uint32 – R/C	40005

EventPtpSyncLocked – Uint32 – R/C 40006

EventSyncIn1Rise – Uint32 – R/C 40010

EventSyncIn1Fall – Uint32 – R/C 40011

EventSyncIn2Rise – Uint32 – R/C 40012

EventSyncIn2Fall – Uint32 – R/C 40013

EventSyncIn3Rise – Uint32 – R/C 40014

EventSyncIn3Fall – Uint32 – R/C 40015

EventSyncIn4Rise – Uint32 – R/C 40016

EventSyncIn4Fall – Uint32 – R/C 40017

EventOverflow – Uint32 – R/C 65534

Always on. Cannot be turned off with EventSelector or EventsEnable1. Event occurs if camera event buffer overflows, i.e. if host is unable to process/send acknowledgements for events as quickly as events are generated from camera.

EventError – Uint32 – R/C 65535

Always on. Cannot be turned off with EventSelector or EventsEnable1. Event should never occur, only returning in case of firmware failure requiring camera repair.

EventNotification – Enum – R/W

Turns the selected event notification *On* or *Off.*

EventSelector – Enum – R/W

Select a specific event to be enabled or disabled using EventNotification. Possible values:

AcquisitionStart
AcquisitionEnd
FrameTrigger
ExposureEnd
AcquisitionRecordTrigger
PtpSyncLost
PtpSyncLocked
SyncIn1Rise
SyncIn1Fall
SyncIn2Rise
SyncIn2Fall
SyncIn3Rise
SyncIn3Fall
SyncIn4Rise

SyncIn4Fall

EventsEnable1 – Uint32 – R/W

Bitmask of all events. Bits correspond to last two digits of EventID. E.g. Bit 1 is *EventAcquisitionStart*, Bit 2 is *EventAcquisitionEnd,* Bit 10 is *EventSyncIn1Rise*. This is an alternative to setting each of the event individually using the *EventNotification* and *EventSelector* method*.*

GigE

BandwidthCtrlMode – Enum – R/W

Select the desired mode of bandwidth control.

StreamBytesPerSecond	The default mode of bandwidth control. See the StreamBytesPerSecond control for more information.
SCPD	Stream channel packet delay expressed in timestamp counter units. This mode is not recommended.
Both	Implements a combination of control modes. This mode is not recommended.

ChunkModeActive – Boolean – R/W

Possible values: *TRUE, FALSE.* Enables camera to send GigE Vision Standard Protocol chunk data with an image. Currently implemented chunk data:

[Bytes 1 – 4] acquisition count.
[Bytes 5 – 8] reserved. 0.
[Bytes 9 – 12] exposure value.
[Bytes 13 – 16] gain value.
[Bytes 17 – 18] sync in levels.
[Bytes 19 – 20] sync out levels.
[Bytes 21 – 24] reserved. 0.
[Bytes 25 – 28] reserved. 0.
[Bytes 29 – 32] reserved. 0.
[Bytes 33 – 36] reserved. 0.
[Bytes 37 – 40] reserved. 0.
[Bytes 41 – 44] chunk ID. 1000.
[Bytes 45 – 48] chunk length.

PvAPI users see tPvFrame.AncillaryBuffer.

NonImagePayloadSize – Unit32 – R/V

Units: Bytes. Size of chunk mode data. If ChunkModeActive = FALSE, NonImagePayloadSize = 0.

PayloadSize – Unit32 – R/V

Units: Bytes. Total size of payload in bytes. Payload = TotalBytesPerFrame + NonImagePayloadSize.

StreamBytesPerSecond – Uint32 – R/W

Range: [1,000,000 – 124,000,000 (248,000,000 for GX in LAG mode]. Units: Bytes/Sec. Moderates the data rate of the camera. This is particularly useful for slowing the camera down so that it can operate over slower links such as Fast Ethernet (100-speed), or wireless networks. It is also an important control for multi-camera situations. When multiple cameras are connected to a single Gigabit Ethernet port (usually through a switch), StreamBytesPerSecond for each camera needs to be set to a value so that the sum of each camera's StreamBytesPerSecond parameter does not exceed the data rate of the GigE port. Setting the parameter in this way will ensure that multiple camera situations work without packet collisions, i.e. data loss.

To calculate the required minimum StreamBytesPerSecond setting for a camera in any image mode, use the following formula:

Height x **Width** x **FrameRate** x **Bytes per Pixel**

115,000,000 is the typical data maximum data rate for a GigE port. Beyond this setting, some network cards will drop packets.

If host reports occasional dropped frames/packets reported as StatFramesDropped/ StatPacketsMissed, with an optimized NIC, you may need to decrease this parameter.

StreamFrameRateConstrain – Boolean – R/W

Possible values: *TRUE, FALSE.* When *TRUE*, camera automatically limits frame rate to bandwidth, determined by StreamBytesPerSecond, to prevent camera buffer overflows and dropped frames. If *FALSE,* frame rate not limited to bandwidth – only sensor readout time. Latter case useful for AcquisitionMode = Recorder, or StreamHoldEnable = On.

StreamHold

For controlling when the camera sends data to the host computer. Normally the camera sends data to the host computer immediately after completion of exposure. Enabling StreamHold delays the transmission of data, storing it in on-camera memory, until StreamHold is disabled.

This feature can be useful to prevent GigE network flooding in situations where a large number of cameras connected to a single host computer are capturing a single event. Using the StreamHold function, each camera will hold the event image data until the host computer disables StreamHold for each camera in turn.

StreamHoldCapacity – Uint32 – R/V

Units: Frames. The total number of images that can be stored in camera memory. Used in AcquisitionMode = Recorder, or StreamHoldEnable = On. Dependent on the camera internal memory size and TotalBytesPerFrame.

StreamHoldEnable – Enum – R/W

Enables StreamHold functionality. When disabled, the image data will be released to the host computer. Possible values: *On, Off.*

Timestamp

TimeStampFrequency – Uint32 – R/C

Units: Hz. Camera clock frequency. Timebase for TimeStampValue. PvAPI users note: images returned from the camera are marked with a timestamp: tPvFrame.TimestampLo/Hi. This can be useful for determining whether images are missing from a sequence due to missing trigger events.

TimeStampReset – Command

Reset the camera's time stamp to 0.

TimeStampValueHi – Uint32 – R/V

Units: Camera clock ticks*2^{32}. Time stamp, upper 32-bit.
TimeStampValueHi*2^{32}/TimeStampFrequency = units in seconds.

TimeStampValueLatch – Command

Command. Latch the value of the timestamp on the camera. Both TimeStampValueHi and TimeStampValueLo are updated with the value read from the camera.

TimeStampValueLo – Uint32 – R/V

Units: Camera clock ticks. Time stamp, lower 32-bit.
TimeStampValueLo/TimeStampFrequency = units in seconds.

ImageFormat

ROI

Region of Interest. Defines a rectangular sub-region of the image. Selecting an ROI that is small can increase the maximum frame rate and reduce the amount of image data. The following parameters define the size and location of the ROI sub-region:

Height – Uint32 – R/W

Range: [1 – cam dependent]. Units: rows. The vertical size of the ROI rectangle.

RegionX – Uint32 – R/W

Range: [0 – cam dependent]. Units: columns. The X position of the top-left corner of the ROI. RegionX + Width must not exceed SensorWidth.

RegionY – Uint32 – R/W

Range: [0 – cam dependent]. Units: rows. The Y position of the top-left corner of the ROI. RegionY + Height must not exceed SensorHeight.

Width – Uint32 – R/W

Range: [1 – cam dependent]. Units: columns. The horizontal size of the ROI rectangle.

PixelFormat – Enum – R/W

The various pixel data formats the camera can output. Not all cameras have every format. See camera user manual.

Pixel Format	**Bit Depth***	**Byte Per Pixel**	**On-Camera Interpolation**	**Description**
Mono8	8	1	Mono Cam: N/A Color Cam: Yes	Mono data.
Mono16	Full	2	N/A	Mono data. Data is LSbit aligned within 16bits. E.g. for 12 bit camera: 0000xxxx xxxxxxxx.
Bayer8	8	1	No	Raw color data.
Bayer16	Full	2	No	Raw color data. Data is LSbit aligned within 16bits. E.g. for 12 bit camera: 0000xxxx xxxxxxxx.
Rgb24	8	3	Yes	Color data. 3 consecutive bytes, R, G, B, per pixel.
Bgr24	8	3	Yes	Color data. 3 consecutive bytes, B, G, R, per pixel.
Yuv411	8	1.5	Yes	Color data. Full Y, limited UV, for 4 pixels extrapolated from 6 bytes.
Yuv422	8	2	Yes	Color data. Full Y, limited UV, for 2 pixels extrapolated from 4 bytes.
Yuv444	8	3	Yes	Color data. Full Y and UV, for 1 pixel extrapolated from 3 bytes.
Rgba32	8	4	Yes	Color data. 4 consecutive bytes, R, G, B, 0, per pixel.
Bgra32	8	4	Yes	Color data. 4 consecutive bytes, B, G, R, 0, per pixel.
Rgb48	Full	6	Yes	Color data. 3 consecutive 16 bit words, R, G, B, per pixel. Data is LSbit aligned within 16bits. E.g. for 12 bit camera: 0000xxxx xxxxxxxx.
Mono12Packed	12	1.5	N/A	Mono data. 2 pixels of data every 3 bytes. Formatted as 11111111, 11112222, 22222222.
Bayer12Packed	12	1.5	No	Raw color data. 2 pixels of data every 3 bytes. Formatted as 11111111, 11112222, 22222222.

*Full bit depth is dependent on the camera A/D. See camera user manual. 8 bit depth = most significant 8 bits of camera A/D.

TotalBytesPerFrame – Uint32 – R/V

Read only. The total number of bytes per image frame. Dependant on ROI, PixelFormat, and Binning.

ImageMode

Binning, i.e. scaling, is the summing of charge of adjacent pixels on a sensor, to give a lower resolution but more sensitive image. For a proportional image, set BinningX and BinningY to equal values.

BinningX – Uint32 – R/W

Range: [cam dependent]. The horizontal binning factor.

BinningY – Uint32 – R/W

Range: [cam dependent]. The vertical binning factor.

Info

CameraName – String – R/W

Human readable camera name. E.g. "EngineRoomCam1".

DeviceFirmwareVersion – String – R/C

Version of the Firmware the camera is running.

DeviceModelName – String – R/W

Human readable model name, such as "GE650". Software should use the *PartNumber* and *PartVersion* to distinguish between models.

DevicePartNumber – String – R/C

Manufacturer's part number

DeviceSerialNumber – String – R/C

The Serial Number is not a unique identifier across models; software should use *UniqueId* instead.

DeviceVendorName – String – R/C

Manufacturer's name.

Firmware

Read only. What firmware is currently loaded on the camera.

FirmwareVerBuild – Uint32 – R/C

Build number.

FirmwareVerMajor – Uint32 – R/C

The major part of the Firmware version number (part before the decimal).

FirmwareVerMinor – Uint32 – R/C

The minor part of Firmware version number (part after the decimal).

Part

PartClass – Uint32 – R/C

Camera part class (manufacturer dependant).

PartNumber – Uint32 – R/C

Camera part number. Manufacturer part number for the camera model.

PartRevision – String – R/C

Camera revision. Part number revision level.

PartVersion – String – R/C

Camera version. Part number version level.

SerialNumber – String – R/C

Camera serial number.

Sensor

SensorBits – Uint32 – R/C

The sensor digitization bit depth.

SensorHeight – Uint32 – R/C

The total number of pixel rows on the sensor.

SensorType – Enum – R/C

Monochrome or Bayer-pattern color sensor type.

SensorWidth – Uint32 – R/C

The total number of pixel columns on the sensor.

UniqueID – Uint32 – R/C

The unique camera ID that differentiates the current camera from all other cameras.

IO

The control and readout of all camera inputs and outputs. The number of inputs and outputs will depend on your camera model.

Strobe

Valid when any of the SyncOut modes are set to Strobe1. Strobe allows the added functionality of duration and delay, useful when trying to sync a camera exposure to an external strobe.

1

Strobe1ControlledDuration – Enum – R/W

Possible Values: *On, Off.* When enabled, the Strobe1Duration control is valid.

Strobe1Delay – Uint32 – R/W

Range [0 – cam dependent]. Units: µs. Delay of start of strobe signal.

Strobe1Duration – Uint32 – R/W

Range [0 – cam dependent]. Units: µs. Duration of strobe signal.

Strobe1Mode – Enum – R/W

Associates the start of strobe signal with one of the following image capture signals:

AcquisitionTriggerReady	Active once the camera has been recognized by the host PC and is ready to start acquisition.
FrameTriggerReady	Active when the camera is in a state that will accept the next frame trigger.

FrameTrigger	Active when an image has been initiated to start. This is a logic trigger internal to the camera, which is initiated by an external trigger or software trigger event.
Exposing	Active for the duration of sensor exposure.
FrameReadout	Active at during frame readout, i.e. the transferring of image data from the CCD to camera memory.
Imaging	Active during exposure and readout.
Acquiring	Active during an acquisition stream.
SyncIn1	Active when there is an external trigger at SyncIn1
SyncIn2	Active when there is an external trigger at SyncIn2

NOTE: Please refer to camera waveform diagrams provided in the camera manuals for more detail information

SyncIn1

SyncIn1GlitchFilter – Uint32 – R/W

Range: [0 – 50000]. Units: ns. Ignores glitches on the SyncIn1 input line with pulse duration less than set value. Note: setting this value increases latency of FrameTrigger by same amount.

SyncIn2

Same as SyncIn1.

SyncInLevels – Uint32 – R/V

A bitmask, each bit corresponding to a specific SyncIn input. For example: 2 equals (0010) which means SyncIn2 is high and all other Sync input signals are low.

SyncOut1

Controls the camera output 1. Can be used for synchronization with other cameras/devices or general purpose outputs.

SyncOut1Invert – Enum – R/W

When enabled, reverses the polarity of the signal output by SyncOut1. Possible values: *On, Off.*

SyncOut1Mode – Enum – R/W

Determines the type of output defined by SyncOut1:

GPO	configured to be a general purpose output, control of which is assigned to SyncOutGpoLevels

AcquisitionTriggerReady	Active once the camera has been recognized by the host PC and is ready to start acquisition.
FrameTriggerReady	Active when the camera is in a state that will accept the next frame trigger.
FrameTrigger	Active when an image has been initiated to start. **NOTE: this pulse is too short to be seen on the camera SyncOut1/2/3/4 output. Do not use!**
Exposing	Active for the duration of sensor exposure.
FrameReadout	Active at during frame readout, i.e. the transferring of image data from the CCD to camera memory.
Acquiring	Active during a acquisition stream.
SyncIn1	Active when there is an external trigger at SyncIn1
SyncIn2	Active when there is an external trigger at SyncIn2
Strobe1	The output signal is controlled according to Strobe1 settings.

Note: Refer to camera waveform diagrams provided in the camera manual for more detailed information.

SyncOut2/3/4

Same as SyncOut1.

SyncOutGpoLevels – Uint32 – R/W

GPO output levels. A bitfield. Bit 0 is sync-out 0, bit 1 is sync-out 1, etc.

Driver Attributes

Driver Attributes affect the host PvAPI driver – vs. Camera Attributes which affect settings on the camera itself. Driver Attributes cannot be saved to a camera ConfigFile.

GigE

Ethernet

DeviceEthAddress – String – R/C

The physical MAC address of the camera.

HostEthAddress – String – R/C

The physical MAC address of the host network card on which the camera is reached.

IP

DeviceIPAddress – String – R/C

The current IP address of the camera.

HostIPAddress – String – R/C

The current IP address of the host network interface.

GvcpRetries – Uint32 – R/W

Gvcp = GigE Vision Control Protocol. The maximum number of resend requests that the host will attempt when trying to recover a lost control packet.

Gvsp

Gvsp = GigE Vision Streaming Protocol.

GvspLookbackWindow – Uint32 – R/W

Units: packets. Size of the look back window when determining if a stream packet is missing.

When a stream packet arrives out of order, the driver skips back GvspLookbackWindow packets to see if the packets previous to this point have all arrived. If not, a resend is issued. A lower value allows the driver less time to assemble out-of-order packets, a larger value allows the driver more time. If the value is set too low, the driver will issue unnecessary resends. If the value is set too high and a packet truly is missing, the driver will issue a resend but the camera may no longer have the required packet in its resend buffer and the packet will be dropped. The ideal value is system dependent.

GvspResendPercent – Float32 – R/W

Range: [1.000 – 100.000] Units: percent. Maximum percentage of missing stream packets in a frame to still generate a driver resend request. Frames with percentage of missing stream packets beyond GvspResendPercent are marked as dropped. Default: 1%.

GvspRetries – Uint32 – R/W

Range: [1 – 100]. Maximum number of resend requests that the host driver will attempt before marking a packet dropped.

GvspSocketBufferCount – Enum – R/W

Possible values: *256,512,1024,2048,4096,8192*. Number of buffers to be used by the network socket. Only applicable when not using the Filter Driver.

GvspTimeout – Uint32 – R/W

Range: [10 – 2500]. Units: ms. Stream packet timeout. If no stream packet received before GvspTimeout, host requests resend, up to GvspRetries times. If still no packet received from camera, packet is marked as dropped.

HeartbeatInterval – Uint32 – R/W

Range: [250 – 3,600,000]. Units: ms. The driver sends heartbeat packets to the camera every HeartbeatInterval milliseconds.

HeartbeatTimeout – Uint32 – R/W

Range: [500 – 3,600,000]. Units: ms. The driver sends heartbeat packets to the camera. If a heartbeat packet is not received within HeartbeatTimeout, the camera assumes the host has closed its controlling application or is dead, and closes its stream and control channel. This parameter may need to be increased if stepping through code in a debugger, as this prevents the driver from sending heartbeat packets.

Multicast

Multicast mode allows the camera to send image data to all hosts on the same subnet as the camera. The host computer that first enables multicast mode is the *master*, and controls all camera parameters. All other hosts / instances are the *monitors*, and can view image data only.

NOTE: Most GigE switches support a maximum PacketSize of 1500 in Multicast mode.

MulticastEnable – Enum – R/W

Possible values: *On, Off.* Enables Multicast mode. Note: In order to enable this, the camera must not be streaming.

MulticastIPAddress – String – R/W

Set the multicast IP address.

PacketSize – Uint32 – R/W

Range: [256 – 32768]. Units: Bytes. Determines the Ethernet packet size. Generally, this number should be set to as large as the network adaptor will allow. If this number is reduced, then CPU loading will increase. Packet sizes > 1500 are called Jumbo Packets/Frames in Ethernet terminology. If your GigE network adaptor does not support Jumbo Packets/Frames of at least 8228 Bytes (the camera default on power up), then you will need to reduce PacketSize parameter to match the maximum supported by your network adaptor.

A PacketSize of 1500 is a safe setting which all GigEthernet network cards support.

Note: If you are seeing all "black images", or all frames reported as StatFramesDropped and zero images reported as StatFramesCompleted, you will likely need to decrease this parameter.

Ptp

Precision Time Protocol (PTP) manages clock synchronization of multiple devices across an Ethernet network. Once the clocks of devices such as cameras, PCs, and sensors are synchronized, future software based triggers can be synchronized within 2 µs. On AVT GigE cameras, the device clock is represented by the camera TimeStampValue attribute.

PtpAcquisitionGateTimeHi – Uint32 – R/W

Range: [0 – (2^{32}-1)] Units: Camera clock ticks*2^{32}. Upper 32 bits of PtpAcquisition trigger time. Must be set beyond current camera TimestampValue. When set, image acquisition stalls. When camera TimestampValue >= PtpAcquisitionGateTime, image acquisition resumes. Used to schedule a synchronized "software trigger" on multiple Ptp synchronized devices.

PtpAcquisitionGateTimeLo – Uint32 – R/W

Range: [0 – (2^{32}-1)] Units: Camera clock ticks. Lower 32 bits of PtpAcquisition trigger time. See PtpAcquisitionGateTimeHi.

PtpMode – Enum – R/W

Off	(Default.) This camera's TimeStampValue is not synchronized with any other device.
Master	This camera's TimeStampValue is the master clock. All other PTP enabled slave devices synchronize their clock to this camera.
Slave	This camera's TimeStampValue is altered to align with a master device's clock.
Auto	This camera uses the IEEE1588 best master clock algorithm to determine which device is master, and which are slaves. It may be assigned as either.

PtpStatus – Enum – R/V

Off	Camera PtpMode set to *Off*.

Master	This camera acting as master clock.
Synching	PTP synchronization not yet achieved. Slave(s) are synching with master.
Slave	PTP synchronization among devices achieved. This camera acting as a slave to another device's master clock.

Stats

StatDriverType – Enum – R/V

Standard	The default network card driver is being used only.
Filter	The AVT filter driver is being used in conjunction with the default network card driver. Using the Filter driver will reduce the load on the host CPU.

StatFilterVersion – String – R/C

Version of the filter driver.

StatFrameRate – Float32 – R/V

Frame rate of the camera.

StatFramesCompleted – Uint32 – R/V

The number of camera images returned to the PvAPI frame queue successfully.

PvAPI programmers note: this stat does not increment if no frames queued.
Use tPvFrame.FrameCount for a counter of exactly which image the camera is returning.

StatFramesDropped – Uint32 – R/V

The number of frames returned to the PvAPI frame queue with one or more dropped packet within.

PvAPI programmers note: this stat does not increment if no frames queued.
Use tPvFrame.FrameCount for a counter of exactly which image the camera is returning.

StatPacketsErroneous – Uint32 – R/V

The number of improperly formed packets. If this number is non-zero, it suggests a possible camera hardware failure.

StatPacketsMissed – Uint32 – R/V

The number of packets missed since the start of imaging.

StatPacketsReceived – Uint32 – R/V

The number of packets received since the start of imaging.

StatPacketsRequested – Uint32 – R/V

The number of resend requests since the start of imaging. When an expected packet is not received by the driver, it is recognized as missing and the driver requests the camera to resend it.

StatPacketResent – Uint32 – R/V

The number of packets resent by the camera and received by the host, since the start of imaging.

Contacting Allied Vision Technologies

- **Technical information:**
http://www.alliedvisiontec.com

- **Support:**
support@alliedvisiontec.com

Allied Vision Technologies GmbH (Headquarters)
Taschenweg 2a
07646 Stadtroda, Germany
Tel.: +49.36428.677-0
Fax.: +49.36428.677-28
e-mail: info@alliedvisiontec.com

Allied Vision Technologies Canada Inc.
101-3750 North Fraser Way
Burnaby, BC, V5J 5E9, Canada
Tel: +1 604-875-8855
Fax: +1 604-875-8856
e-mail: info@alliedvisiontec.com

Allied Vision Technologies Inc.
38 Washington Street
Newburyport, MA 01950, USA
Toll Free number +1-877-USA-1394
Tel.: +1 978-225-2030
Fax: +1 978-225-2029
e-mail: info@alliedvisiontec.com