Allied Vision GigE Cameras

Camera and Driver Attributes

Bigeye G Firmware v3.1.44.6
Mako G Firmware v1.54
Manta Firmware v1.54
Prosilica Firmware v1.54

V1.3.1

20 March 2015

Allied Vision Technologies GmbH
Taschenweg 2a
D-07646 Stadtroda, Germany

ALLIED Vision Technologies

Legal notice

Trademarks

Unless stated otherwise, all trademarks appearing in this document of Allied Vision Technologies are brands protected by law.

Warranty

The information provided by Allied Vision Technologies is supplied without any guarantees or warranty whatsoever, be it specific or implicit. Also, excluded are all implicit warranties concerning the negotiability, the suitability for specific applications or the non-breaking of laws and patents. Even if we assume that the information supplied to us is accurate, errors and inaccuracy may still occur.

Copyright

All texts, pictures and graphics are protected by copyright and other laws protecting intellectual property. It is not permitted to copy or modify them for trade use or transfer, nor may they be used on websites.

Managing Director: Mr. Frank Grube
Tax ID: DE 184383113

Headquarters:

Taschenweg 2a
D-07646 Stadtroda, Germany
Tel: +49 (0)36428 6770
Fax: +49 (0)36428 677-28
e-mail: info@alliedvision.com

Contents

Contacting Allied Vision Technologies

Info

- **Technical information:**
  http://www.alliedvision.com
- **Support:**
  support@alliedvision.com

Allied Vision Technologies GmbH (Headquarters)
Taschenweg 2a
07646 Stadtroda, Germany
Tel.: +49 36428-677-0
Fax: +49 36428-677-28
e-mail: info@alliedvision.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@alliedvision.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@alliedvision.com

Allied Vision Technologies Asia Pte. Ltd.
82 Playfair Road
#07-02 D'Lithium
Singapore 368001
Tel.: +65 6634-9027
Fax: +65 6634-9029
e-mail: info@alliedvision.com

Allied Vision Technologies (Shanghai) Co., Ltd.
2-2109 Hongwell International Plaza
1602# ZhongShanXi Road
Shanghai 200235, China
Tel.: +86 (21) 64861133
Fax: +86 (21) 54233670
e-mail: info@alliedvision.com

Introduction

The document describes the standard and advanced camera controls for AVT GigE cameras as seen from the AVT GigE SampleViewer. The document is intended for use with **PvAPI SDK**. AVT offers a number of GigE Vision camera families, which includes:

- Bigeye G
- Mako G
- Manta
- Prosilica GB
- Prosilica GC
- Prosilica GE
- Prosilica GS
- Prosilica GT
- Prosilica GX

This document can be applied to all of these families.

www

Follow this link to learn about GigE Vision cameras from AVT.

http://www.alliedvision.com/en/products/cameras

Important notes

Note

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.

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

Note

Uint32 and Float32 ranges: where camera dependent, see camera user manual, or see slider control in **AVT GigE SampleViewer**. PvAPI users see *PvAttrRangeUint32*, *PvAttrRangeFloat32* calls.

Note

- R/W = attribute is read/write
- R/C = attribute is read only and constant
- R = attribute is read only and may change at any time

Document history

Version	Date	Remarks
V1.0.0	2006-May-18	New Manual – Release Status – Firmware: 1.00.00
V1.0.1	2006-Jun-12	Firmware: 1.14.00 – ExposureMode, WhitebalMode addition
V1.0.2	2006-Aug-02	Firmware: 1.18.00 – PixelFormat YUV addition
V1.0.3	2006-Sep-08	Firmware: 1.22.00 – StreamHold, SyncOutGPOLevels addition
V1.0.4	2007-May-30	Firmware: 1.26.00 – Iris, AcquisitionMode, StreamBytesPerSecond, StreamHoldCapacity addition
V1.0.5	2010-Feb-10	Firmware: 1.38.00 – EventControls, GVSP addition
V1.0.6	2010-Feb-23	Firmware: 1.40.00 – LensDrive, DefectMaskColumnEnable, ChunkModeActive addition
V1.0.7	2010-Nov-02	Firmware: 1.42.00 – StreamFrameRateConstrain, FrameStartTriggerOverlap, SyncIn1GlitchFilter addition – Note on auto exposure plus auto gain priority added
V1.0.8	2012-Feb-20	Firmware: 1.48.01 – PTP, LensDCIris, LensPIris, DeviceTemperatureMainboard addition
V1.0.9	2013-Jan-14	Firmware: 1.50.01 – DeviceTemperatureSensor addition – FrameTrigger removed from SyncOutMode – DSPSubregion upper limits changed from 4294967295 to sensor limit – Added Manta camera controls: LUTControl, Offset, Decimation, NirMode
V1.1.0	2013-Jul-05	• Added Bigeye G camera controls • Added Mako G controls • Added contact information for Allied Vision Technologies (Shanghai) Co. Ltd. • Changed user access from R/V to R
		to be continued on next page

Table 1: Document History

continued from last page		
V1.1.0 [Continued]	2013-Jul-05 [Continued]	• Updated the following controls: – PayloadSize – EdgeFilter – Gamma – Hue – IrisVideoLevelMax – IrisVideoLevelMin – Saturation – LUTControl – BandwidthCtrlMode – StreamHoldEnable
V1.1.1	2013-Sep-06	• Added the EF lens controls • Added control on page 48 • Updated the DefectMaskPixelEnable, Eventcontrol, and DeviceStatus controls
V1.1.2	2014-Oct-08	• Merged camera controls and driver controls chapters • Added Index and Legal notice • Updated HeartbeatInterval, HeartbeatTimeout, GvcpRetries, EventID, and ChunkModeActive control • Updated BinningX, BinningY, DecimationHorizontal, and DecimationVertical controls • Updated ExposureAutoOutliers, ExposureValue, Gain, GainAutoMax, GainAutoMin, and GainAutoOutliers • Removed FrameTrigger from SyncOut1Mode • Moved ReverseX under ImageMode category • Added ReverseY and ExposureTimeIncrement • Updated PTP and TimeStampReset
V1.2.0	2015-Jan-15	• Added SensorShutterMode, BinningVerticalMode, BinningHorizontalMode, and DefectMaskEnable • Updated BinningX and BinningY • Added PieceWiseLinearHDR option in ExposureMode • Added ExposureValuePWL1, ExposureValuePWL1, ThresholdPWL1, and ThresholdPWL1 • Updated ExposureValue, FrameRate, GainMode, IrisMode, and WhitebalMode
V1.3.0	2015-Mar-10	• Added EFLensControl • Updated DefectMaskEnable, ChunkModeActive, PtpStatus, PtpMode, and SensorShutterMode
V1.3.1	2015-Mar-20	• Replaced old links with new Allied Vision website links

Table 1: Document History

Symbols used in this manual

Note This symbol highlights important information.

www This symbol highlights URLs for further information. The URL itself is shown in blue.

Example:

http://www.alliedvision.com

Additional information

AVT software

All software packages provided by AVT are **free of charge** and contain the following components:

- Drivers
- Software Development Kit (SDK) for camera control and image acquisition
- Examples based on the provided APIs of the SDK
- Documentation and release notes
- Viewer application to operate/configure the cameras

www All **software packages** (including **documentation** and **release notes**) provided by AVT can be downloaded at:

http://www.alliedvision.com/en/support/software-downloads

AVT GigE camera attributes

Acquisition

Trigger

AcqEnd

AcqEndTriggerEvent – Enum – R/W

If ***AcqEndTriggerMode*** = *SyncIn1/2/3/4*, determines which *SyncIn* electrical signal initiates trigger.

EdgeRising	[Default] Rising edge trigger
EdgeFalling	Falling edge trigger
EdgeAny	Rising or falling edge
LevelHigh	Active high signal
LevelLow	Active low signal

AcqEndTriggerMode – Enum – R/W

Determines if 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
SyncIn3	Trigger at *SyncIn3* to be associated with this control
SyncIn4	Trigger at *SyncIn4* to be associated with this control
Disabled	[Default] 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 an ***AcqRec*** trigger. See ***AcquisitionMode*** = *Recorder*.

AcqRecTriggerEvent – Enum – R/W

If ***AcqRecTriggerMode*** = *SyncIn1/2/3/4*, determines which *SyncIn* electrical signal initiates trigger.

EdgeRising	[Default] Rising edge trigger
EdgeFalling	Falling edge trigger
EdgeAny	Rising or falling edge
LevelHigh	Active high signal
LevelLow	Active low signal

AcqRecTriggerMode – Enum – R/W

Determines if recorder mode trigger event is initiated by an external hardware trigger.

SyncIn1	[Default] Trigger at *SyncIn1* to be associated with this control
SyncIn2	Trigger at *SyncIn2* to be associated with this control
SyncIn3	Trigger at *SyncIn3* to be associated with this control
SyncIn4	Trigger at *SyncIn4* 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

AcqStart controls relate to triggering the start of an acquisition stream. Frames are triggered within this acquisition stream. See ***FrameStart*** for triggering frames.

AcqStartTriggerEvent – Enum – R/W

If ***AcqStartTriggerMode*** = *SyncIn1/2/3/4*, determines which *SyncIn* electrical signal initiates trigger.

EdgeRising	[Default] Rising edge trigger
EdgeFalling	Falling edge trigger
EdgeAny	Rising or falling edge
LevelHigh	Active high signal
LevelLow	Active low signal

AcqStartTriggerMode – Enum – R/W

Determines if 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
SyncIn3	Trigger at *SyncIn3* to be associated with this control
SyncIn4	Trigger at *SyncIn4* to be associated with this control
Disabled	[Default] No external trigger. Acquisition must be started with the ***AcquisitionStart*** API command

FrameRate – Float32 – R/W

Range: [Camera dependent] Units: Hz

When ***FrameStartTriggerMode*** is set to *FixedRate*, this control specifies the frame rate. Depending on the exposure duration, the camera may not achieve the frame rate set here.

Note

- If ***ExposureMode*** = *Manual*:
  Ensure [1/***ExposureValue****] > ***FrameRate*** to achieve target frame rate.
- If ***ExposureMode*** = *External*:
  Ensure [1/(external trigger pulse width)] > ***FrameRate*** to achieve target frame rate.
- If ***ExposureMode*** = *PieceWiseLinearHDR*:
  Ensure the [1/***ExposureValue****] > ***FrameRate*** to achieve target frame rate.

* ***ExposureValue*** in seconds

FrameStart

FrameStart controls relate to triggering individual frames within an acquisition stream. See ***AcqStart*** for triggering an acquisition stream.

FrameStartTriggerDelay – Uint32 – R/W

Range: [0 - Camera dependent] Default: *0* Units: µs

Start of frame is delayed ***FrameStartTriggerDelay*** µs 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 sync with a strobe lighting source, which will have some fixed setup time.

FrameStartTriggerEvent – Enum – R/W

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

EdgeRising	[Default] 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 a frame is initiated.

Note

An acquisition stream must be started in order to trigger/receive individual frames. For *Freerun* and *FixedRate* the first frame is synchronized to ***AcquisitionStart***/***AcqStart*** trigger.

Freerun	[Default] 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*
SyncIn3	External trigger *SyncIn3*
SyncIn4	External trigger *SyncIn4*
FixedRate	Frame triggers generated on-camera, at frame rate defined by *FrameRate* attribute
Software	Software initiated frame trigger. See ***FrameStartTrigger-Software*** command

FrameStartTriggerOverlap – Enum – R/W

Does not work with Software triggering. Only external.

Off	[Default] When *Off*, any external trigger received before ***FrameTriggerReady*** signal is high is ignored
PreviousFrame	When *PreviousFrame*, any external trigger received before ***FrameTriggerReady*** is latched and used to trigger the next frame

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] Default: *1* 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	[Default] 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/ acquiring 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] Default: *0* 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*** = *Recorder*.

Note

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

SensorShutterMode – Enum – R/W

Type of the shutter. Figure 1 illustrates different sensor shutter modes.

Global	[Default] All pixels reset and start exposure at same time. All pixels shifted to readout at same time. All pixels have same ***ExposureValue***
Rolling	Each row is reset, exposed, and read out in succession from top to bottom of image. All pixels have same ***ExposureValue***. This mode is susceptible to motion blur; however, this mode offers enhanced SNR/dynamic range
GlobalReset	All pixels reset and start exposure at same time. Pixels are shifted to readout one line at a time from top to bottom of image. This mode does not allow overlapped exposure and readout. In this mode, ***ExposureValue*** is the time from global reset to start of readout of top row. Subsequent rows will have a longer exposure time (***ExposureValue*** + row readout time * row number). This mode offers enhanced SNR/dynamic range with no motion blur, which is useful for strobe applications

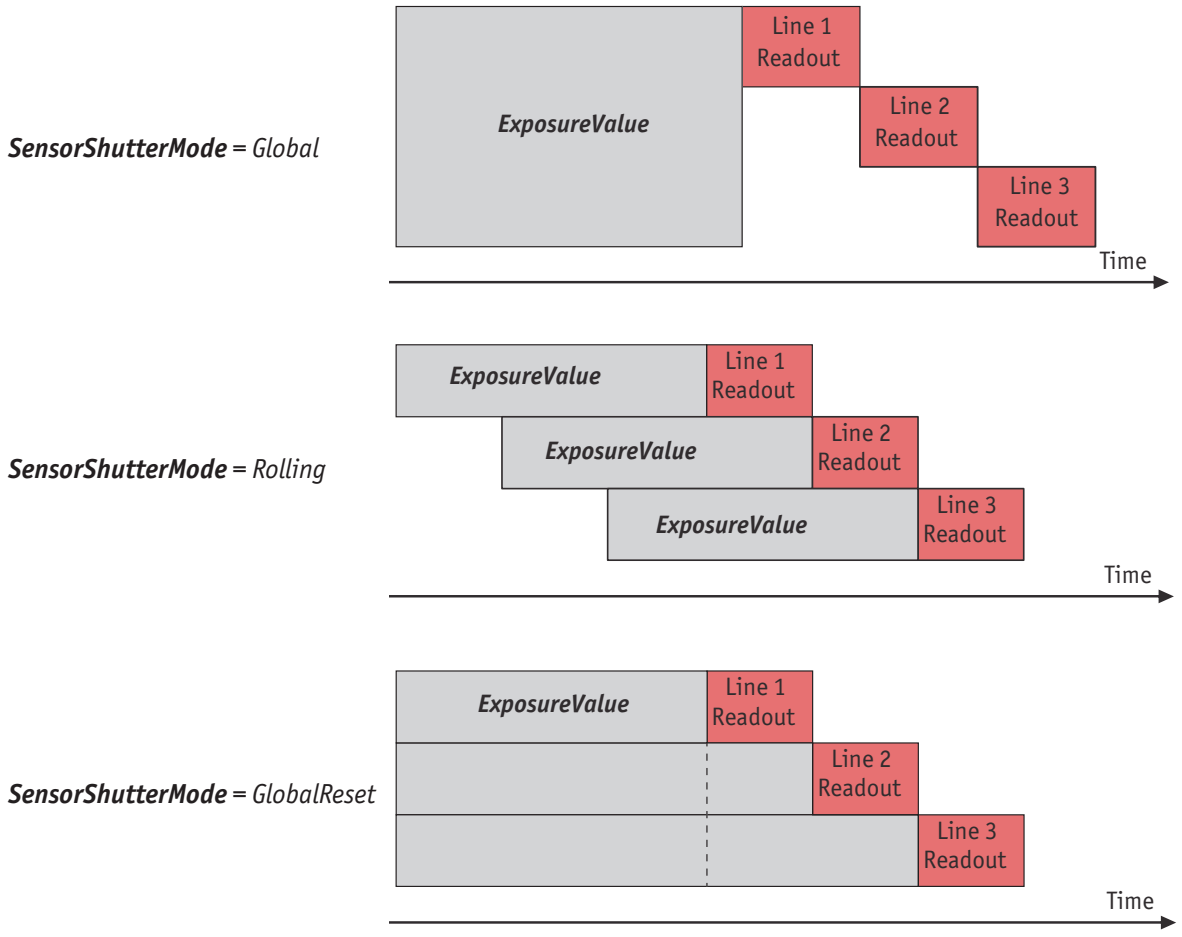

Figure 1: Illustration showing different sensor shutter modes

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 To save the content of a LUT, use ***LUTSave*** or ***LUTSaveAll***

ConfigFileIndex – Enum – R/W

Possible values: *Factory, 1, 2, 3, 4, 5* Default: *Factory*

Index number corresponds 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* Default: *Factory*

Saved configuration is 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

The ***ColorTransformationControl*** section describes features related to color transformations in the AVT GigE color cameras.

Definition. The **color transformation** is a linear operation taking as input the triplet R_{in}, G_{in}, B_{in} for an RGB color pixel. This triplet is multiplied by a 3x3 matrix.This color transformation allows changing the coefficients of the 3x3 matrix.

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

See ColorTransformationValue## attributes.

ColorTransformationMode – Enum – R/W

Off	No color transformation
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] Default: *1.000*

Blue multiplicative factor applied to blue input channel.

ColorTransformationValueBG – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Green multiplicative factor applied to blue input channel.

ColorTransformationValueBR – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Red multiplicative factor applied to blue input channel.

ColorTransformationValueGB – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Blue multiplicative factor applied to green input channel.

ColorTransformationValueGG – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Green multiplicative factor applied to green input channel.

ColorTransformationValueGR – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Red multiplicative factor applied to green input channel.

ColorTransformationValueRB – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Blue multiplicative factor applied to red input channel.

ColorTransformationValueRG – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Green multiplicative factor applied to red input channel.

ColorTransformationValueRR – Float32 – R/W

Range: [0.000 – 2.000] Default: *1.000*

Red multiplicative factor applied to red input channel.

DSP

The automatic exposure, gain, white balance, 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 – Sensor height] Default: *Sensor height*

Defines the bottom edge of the DSP subregion. Units: Rows from top edge of full image.

DSPSubregionLeft – Uint32 – R/W

Range: [0 – Sensor width] Default: *0*

Defines the left edge of the DSP subregion. Units: Columns from left edge of full image.

DSPSubregionRight – Uint32 – R/W

Range: [0 – Sensor width] Default: *Sensor width*

Defines the right edge of the DSP subregion. Units: Columns from left edge of full image.

DSPSubregionTop – Uint32 – R/W

Range: [0 – Sensor height] Default: *0*

Defines the top edge of the DSP subregion. Units: Rows from top edge of full image.

DefectMaskEnable – Boolean – R/W

Enables or disables masking of defective pixel. Defective pixels are replaced with averaged values from neighboring pixels.

True	[Default] Enables defect masking
False	Disables defect masking

Note

If ***BinningX***, ***BinningY***, ***DecimationHorizontal***, or ***DecimationVertical*** is set greater than 1, ***DefectMaskEnable*** is set to *False*.

www

For more information on the Defect Mask Loader and defect masking process, see:

http://www.alliedvision.com/fileadmin/content/documents/products/cameras/various/appnote/Defect_Masking.pdf

DefectMask

Some larger format sensors may contain defective columns. Class 1 and Class 0 sensors are available with no defective columns.

www

See the **Modular Concept** document, or contact your AVT sales representative for more information:
http://www.alliedvision.com/fileadmin/content/documents/products/cameras/various/modular-concept/Modular_concept_external.pdf

DefectMaskColumnEnable – Enum – R/W

Defect masking replaces defective columns with interpolated values based on neighboring columns. Defective columns are detected and recorded at the factory.

Enabled	[Default] Enables masking of defective columns
Disabled	Disables masking of defective columns

www

For more information on the **Loaddefect** application and column defect masking process, see:

http://www.alliedvision.com/fileadmin/content/documents/products/cameras/various/appnote/Column_Defect_Masking.pdf

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

Note

EdgeFilter feature is applicable only to color models/Manta cameras except Manta type B camera models.

EFLensControl

The section describes features related to EF lens control in the GigE cameras with integrated EF-Mount.

Note

The features listed under ***EFLensControl*** are NOT available for cameras with Birger EF-Mount option.

EFLensFStop

EFLensFStopCurrent – Float – R/W

Range: [***EFLensFStopMin*** - ***EFLensFStopMax***]
Current F-stop number or aperture of the EF lens.

EFLensFStopDecrease – Command

Decrease F-stop number, i. e., increase lens aperture by the ***EFLensFStopStepSize***.

EFLensFStopIncrease – Command

Increase F-stop number, i. e., reduce lens aperture by the ***EFLensFStopStepSize***.

EFLensFStopMax – Float – R

Default: Lens dependent Unit: F-Stop
Maximum possible F-stop setting or the smallest possible aperture for the EF lens based on current zoom setting.

EFLensFStopMin – Float – R

Default: Lens dependent Unit: F-Stop
Minimum possible F-stop setting or the largest possible aperture for the EF lens based on current zoom setting.

EFLensFStopStepSize – Integer – R/W

Range: [1–8] Units: F-Stop/8
Size of increments/decrements in ***EFLensFStopCurrent*** when using ***EFLensFStopIncrease*** and ***EFLensFStopDecrease*** commands, respectively.

EFLensFocus

EFLensFocusCurrent – Integer – R/W

Range: [***EFLensFocusMin*** – ***EFLensFocusMax***]
Current focus setting.

EFLensFocusDecrease – Command

Decrease/shorten focus distance by ***EFLensFocusStepSize***.

EFLensFocusIncrease – Command

Increase/lengthen focus distance by ***EFLensFocusStepSize***.

EFLensFocusMax – Integer – R

Default: Lens dependent
Maximum/farthest possible focus setting.

EFLensFocusMin – Integer – R

Default: Lens dependent
Minimum/nearest possible focus setting.

EFLensFocusStepSize – Integer – R/W

Range: [1 – Lens dependent] Default: *10*
Size of increments/decrements in ***EFLensFocusCurrent*** when using ***EFLensFocusIncrease*** and ***EFLensFocusDecrease*** commands, respectively.

EFLensFocusSwitch – Enum – R

Current position of lens AF/MF switch.

AutoFocus	Switch is in auto focus (AF) position
ManualFocus	Switch is in manual focus (MF) position

Note All controls under ***EFLensFocus*** become read-only when the lens AF/MF switch is set to manual focus (MF).

EFLensStatus

EFLensID – Integer – R
Identification value of the attached EF lens.

EFLensLastError – Enum – R
Most recently detected error.

EFLensErrNone	No error detected
EFLensErrQuery	Lens failed query by camera
EFLensErrInternal1	Lens communication error (can occur when removing lens)
EFLensErrInternal2	Lens communication error (can occur when removing lens)
EFLensErrBusy	Lens remained busy for longer than 10 seconds
EFLensErrZeroStop	Lens focus “Zero Stop” not detected
EFLensErrInfinityStop	Lens focus “Infinity Stop” not detected

EFLensState – Enum – R
Current EF lens state.

EFLensIdle	No lens action in progress
EFLensBusy	Lens is busy (changing focus or aperture)
EFLensWaiting	Camera is waiting for lens attachment
EFLensInitializing	Camera is initializing lens
EFLensError	Lens Error detected. Error type is indicated by ***EFLensLastError***. Remains in this state until ***EFLensInitialize*** is executed

EFLensInitialize – Command
Initializes the EF lens. This command is automatically executed on power up and/or when lens is attached to camera.

EFLensZoom

EFLensZoomCurrent – Integer – R
Range: [***EFLensZoomMin*** – ***EFLensZoomMax***] Units: mm
Current focal length of the EF lens.

EFLensZoomMax – Integer – R
Default: Lens dependent Units: mm
Maximum focal length of the EF lens.

EFLensZoomMin – Integer – R
Default: Lens dependent Units: mm
Minimum focal length of the EF lens.

Exposure

Auto

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.

Note

The camera must be acquiring images in order for the auto algorithm to update.

ExposureAutoAdjustTol – Uint32 – R/W

Range: [0 – 50] Default: *5* 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	[Default] 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: [Camera dependent] Default: *500000* 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: [Camera dependent] Default: *Camera dependent* Units: µs

The lower bound to the exposure setting in *autoexposure* mode.

ExposureAutoOutliers – Uint32 – R/W

Range: [0 – 1000] Default: *0* Units: 0.01% i.e. 1000 = 10%

With ***ExposureAutoTarget*** as the mean target brightness, ***ExposureAutoOutliers*** is the percentage of pixels on the upper bound of the image brightness distribution graph that are ignored by the *ExposureAuto* algorithm. This can be used limit the effect of small specular bright spots on the overall image brightness calculation.

ExposureAutoRate – Uint32 – R/W

Range: [1 – 100] Default: *100* Units: percent

The rate at which the auto exposure function changes the exposure setting.

ExposureAutoTarget – Uint32 – R/W

Range: [0 – 100] Default: *50* 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	[Default] 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*
PieceWiseLinearHDR	Image dynamic range is increased in difficult lighting situations by clamping down bright pixels with light levels beyond ***ThresholdPWL*** limits. Overall, camera exposure time is set by ***ExposureValue***. HDR sub-exposures are set using ***ExposureValuePWL1*** and ***ExposureValuePWL2***

ExposureTimeIncrement – Float32 – R/C

Range: [Camera dependent] Units: µs

Increment/resolution of the exposure time in microseconds.

ExposureValue – Uint32 – R/W

Range: [Camera dependent] Units: µs

The sensor integration time. Values written to control are rounded to nearest multiple of ***ExposureTimeIncrement***. Reading this control returns the used, rounded value.

ExposureValue depends on ***ExposureMode*** as follows:

- ***ExposureMode*** = *Manual*: ***ExposureValue*** is sensor integration time.
- ***ExposureMode*** = *Auto/AutoOnce/External*: ***ExposureValue*** is ignored.
- ***ExposureMode*** = *PieceWiseLinearHDR*: ***ExposureValue*** is the full sensor integration time. See ***ExposureValuePWL1*** and ***ExposureValuePWL2*** for setting ***ThresholdPWL*** exposure durations.

ExposureValuePWL1 – Float – R/W

Range: [Camera dependent] Unit: µs

Valid only when ***ExposureMode*** = *PieceWiseLinearHDR*. Exposure time after ***ThresholdPWL1*** is reached.

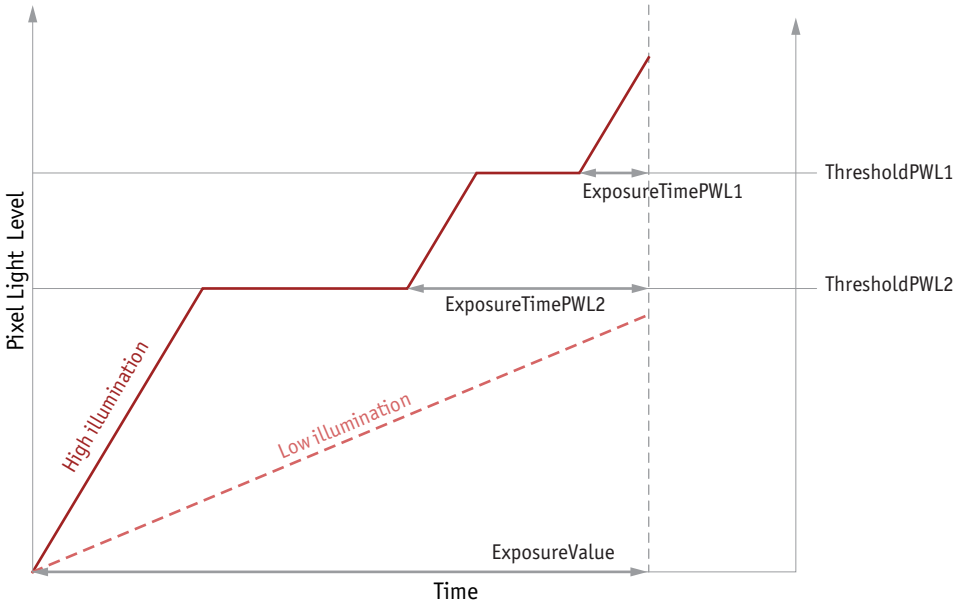

Figure 2: HDR sub exposures and thresholds when ***ExposureMode*** = *PieceWiseLinearHDR*

ExposureValuePWL2 – Float – R/W

Range: [Camera dependent] Unit: µs

Valid only when ***ExposureMode*** = *PieceWiseLinearHDR*. Exposure time after ***ThresholdPWL2*** is reached.

Note

When ***ThresholdPWL2*** is less than ***ThresholdPWL1*** (i.e. enabled), ***ExposureValuePWL2*** must be greater than ***ExposureValuePWL1***.

ThresholdPWL1 – Integer – R/W

Range: [0–63] Default: *63*

Valid only when ***ExposureMode*** = *PieceWiseLinearHDR*. The first and highest threshold level in *PieceWiseLinearHDR*. *0* = no light capacity, *63* = full pixel light capacity.

Note

Leaving ***ThresholdPWL1*** at 63 disables the first threshold of *PieceWiseLinearHDR* mode, effectively disabling HDR mode.

ThresholdPWL2 – Integer – R/W

Range: [0–63] Default: *63*

Valid only when ***ExposureMode*** = *PieceWiseLinearHDR*. The second and lowest threshold level in *PieceWiseLinearHDR*. *0* = no light capacity, *63* = full pixel light capacity.

Note

Setting ***ThresholdPWL2*** above ***ThresholdPWL1*** disables the second threshold of ***PieceWiseLinearHDR*** mode.

Shutter – Enum – R/W

Activate or deactivate the mechanical shutter of Bigeye G-629B Cool cameras.

Off	Deactivate the mechanical shutter. Use this mode, if you operate the camera with pulsed light sources
On	[Default] Activate the mechanical shutter. If activated, the mechanical shutter opens upon each exposure cycle and closes again, when the exposure is over. Use this mode, if you operate the camera with constant light sources, due to the full frame sensor
SyncIn1	Enables or disables the mechanical shutter dependent on the level of *SyncIn1*
SyncIn2	Enables or disables the mechanical shutter dependent on the level of *SyncIn2*
SyncIn3	Enables or disables the mechanical shutter dependent on the level of *SyncIn3*
SyncIn4	Enables or disables the mechanical shutter dependent on the level of *SyncIn4*
SyncIn5	Enables or disables the mechanical shutter dependent on the level of *SyncIn5*

Note

The shutter feature is intended to control the exposure by means of a mechanical shutter. It should not be confused with any other exposure control feature.

The mechanical shutter is available **ONLY** on the Bigeye G-629B Cool camera.

Gain

Auto

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. The camera must be acquiring images in order for the auto algorithm to update.

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] Default: *5* Units: percent

Tolerance in variation from ***GainAutoTarget*** in which the auto exposure algorithm will not respond.This attribute is used to limit auto gain changes to only larger variations in scene lighting.

GainAutoMax – Uint32 – R/W

Range: [0 – Camera dependent] Units: [1, 0.1 dB camera dependent]

The upper bound to the gain setting in auto gain mode.

GainAutoMin – Uint32 – R/W

Range: [0 – Camera dependent] Default: *0* Units: [1, 0.1 dB camera dependent]

The lower bound to the gain setting in Auto gain mode. Normally this number would be set to zero.

GainAutoOutliers – Uint32 – R/W

Range: [1 – 1000] Default: *0* Units: 0.01%, i.e., 1000 = 10%

With ***GainAutoTarget*** as the mean target brightness, ***GainAutoOutliers*** is the percentage of pixels on the upper bound of the image brightness distribution graph that are ignored by the *GainAuto* algorithm. This can be used limit the effect of small specular bright spots on the overall image brightness calculation.

GainAutoRate – Uint32 – R/W

Range: [1 – 100] Default: *100* Units: percent

The rate at which the auto gain function changes. A percentage of the maximum rate.

GainAutoTarget – Uint32 – R/W

Range: [0 – 100] Default: *50* Units: percent

The general lightness or darkness of the auto gain feature. A percentage of maximum ***GainValue***.

GainMode – Enum – R/W

Manual	[Default] The camera gain is fixed by ***GainValue*** parameter
Auto	Valid when ***ExposureMode*** = *Manual*, *Auto*, or *PieceWiseLinear-HDR*. The gain will vary continuously according to the scene illumination. The *Auto* function operates according to the Auto and DSP controls
AutoOnce	Valid when ***ExposureMode*** = *Manual*, *Auto*, or *PieceWiseLinear-HDR*. 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
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*

GainValue – Uint32 – R/W

Range: [Camera dependent] Default: *0* Units: [1, 0.1 dB camera dependent]

$$G_{dB} = 20\log\left(\frac{V_{out}}{V_{in}}\right)$$

This is the gain setting applied to the sensor. For best image quality, the gain setting should be set to zero. However, in low-light situations, it may be necessary to increase the gain setting.

Gamma – Float32 – R/W

Range: [Camera dependent] Default: *1.000* Units: Output = $(\text{Input})^{\text{Gamma}}$

Nonlinear brightness control.

Hue – Float32 – R/W

Range: [Camera dependent] Default: *0.00* 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.

IODMode - Enum - R/W

Set camera to continuous or Image on Demand (IOD) mode.

Continuous	The camera requires no external exposure signal. The camera generates a constant exposure time independently. The exposure time is equal to frame readout time and cannot be adjusted. Bigeye G-132B Cool and Bigeye G-283B Cool achieve maximum frame rate in *Continuous* mode only.

IOD	[Default] Enables *IOD* mode (image on demand mode). In this mode the camera needs an external trigger signal or a timer driven internal exposure signal
SyncIn1/2/3/4/5	The camera is switched between *IOD* and *Continuous* mode, dependent on the level of *SyncIn1/2/3/4/5*

Note

If *Continuous* mode is activated, no external exposure signal is allowed. Set e.g. ***FrameStartTriggerMode*** to an unused *SyncIn*.

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 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

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 dependent): video, DC, and 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 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/PIris-Auto* results in undefined, "race to target" behavior.

Note The camera must be acquiring images in order for the auto algorithm to update.

IrisAutoTarget – Uint32 – R/W

Range [0 – 100] Default: *50* 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. Valid when ***ExposureMode*** = *Manual* or *PieceWiseLinear-HDR*.

Disabled	[Default] 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-Iris auto mode. 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

Dependant on lens type:

Lens type	Range	Description
Video-type lenses	[0 – 150] Units: 10 mV	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	[0-100]	Attempts to match ***IrisAutoTarget***
DC-iris lenses	[0-100]	Attempts to match ***IrisAutoTarget***

IrisVideoLevelMax – Uint32 – R/W

Range: [0 – 150] Default: *Camera dependent* Units: 10 mV [Manta: 13.2 mV]

Video-type lenses only. Limits the maximum driving voltage for closing the lens iris.

IrisVideoLevelMin – Uint32 – R/W

Range: [0 – 150] Default: *Camera dependent* Units: 10 mV [Manta: 13.2 mV]

Video-type lenses only. Limits the minimum driving voltage for opening the lens iris.

LensDCIris

DC Iris lenses only.

LensDCDriveStrength – Uint32 – R/W

Range: [0 – 50] Default: 10

Lens drive voltage. Altering this changes the speed at which a DC-Iris lens operates. The lower the value, the slower the lens operates. A higher value may result in iris oscillation. The optimum value is lens dependent. 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.

www

For a list of P-Iris supported lenses, along with their ***LensPIrisFrequency*** and ***LensPIrisNumSteps*** specifications:

http://www.alliedvision.com/fileadmin/content/documents/products/cameras/various/appnote/P-iris_Lenses_Supported_by_Prosilica_GT_Cameras.pdf

LensPIrisFrequency – Uint32 – R/W

Range: [0 – 1000] Default: 100 Units: Hz

Stepping motor drive rate. Lens dependent. Use value defined in application note on supported P-iris lenses or contact lens manufacturer.

LensPIrisNumSteps – Uint32 – R/W

Range: [1 – 1023] Default: 50

Maximum number of discrete iris/aperture positions. Use value defined in application note on supported P-iris lenses, or contact lens manufacturer.

LensPIrisPosition – Uint32 – R/W

Range: [0 – 1022] Default: 50

Iris/aperture position. Manually control iris in ***PIrisManual*** mode, or read 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. Applied post-bayer interpolation. Only available on color ***PixelFormats*** noted with on-camera interpolation.

0.000	Monochrome
1.000	[Default] Default saturation
2.000	Maximum possible saturation that can be applied

LUTControl

The use of one LUT allows any function (in the form Output = F(Input)) to be stored in the camera's memory and to be applied on the individual pixels of an image at runtime.

Note **Color cameras only:**

LUTControl with single color panes will not work when binning is enabled, due to loss of color information.

LUTInfo

This control provides active LUT information.

LUTAddress – Integer – R/C

Indicates location of memory when LUT is loaded.

LUTSizeBytes – Integer – R/C

Size of the memory area where the LUT is located.

LUTBitDepthIn – Integer – R/C

Bit depth of the input value of the LUT block.

LUTBitDepthOut – Integer – R/C

Bit depth of the output value of the LUT block.

LUTEnable – Boolean – R/W

Possible values: True, False Default: *False*

Activates or deactivates the selected LUT.

LUTIndex – Integer – R/W

Range: [0 – ($2^{LUTBitDepthIn}$ - 1)] Default: *0*

Controls the index (offset) of the coefficient to access in the selected LUT.

LUTLoad/LUTLoadAll – Command

Loads LUT from flash memory into volatile memory of the camera.

LUTMode – Enum – R/W

Selects on which pixels the selected LUT will be applied.

Luminance	[Default] LUT is applied on all pixels
Red	LUT is applied on red pixels only
Green	LUT is applied on green pixels only
Blue	LUT is applied on blue pixels only

Note

To avoid confusion, especially with color cameras, we recommend the following steps:

1. Configure the LUT modes.
2. Enable the LUT.

LUTSave/LUTSaveAll – Command

Saves LUT from volatile memory into flash memory of the camera.

Note

With **ConfigFile** control (***ConfigFileSave*** command) you can't save the contents of the LUT.

LUTSelector – Enum – R/W

Possible values: LUT1, LUT2, LUT3, LUT4, LUT5 Default: *LUT1*

Selects which LUT to control. These LUTs are camera specific.

LUTValue – Integer – R/W

Range: [0 – ($2^{LUTBitDepthOut} - 1$)] Default: *4095*

Returns or sets the value at entry ***LUTIndex***.

NirMode – Enum – R/W

Manta NIR models only.

Selects the NIR modes. These modes differ in quantum efficiency, frame rates, and anti-blooming characteristics.

Off	NirMode set off. Acquire and readout image at same time: • **NIR sensitivity:** No increased sensitivity in NIR range • **Anti-blooming characteristics:** As specified by sensor manufacturer • **Usage:** Best suited if you need very long exposure time
On_HighQuality	[Default] Can't acquire and readout image at same time. The exposure time will always influence frame rate directly: • **NIR sensitivity:** Increased NIR sensitivity, except for a very small portion of the exposure time, which is: $t_{NormalQE}$ = MIN(4300 µs, ExposureValue/4) • **Anti-blooming characteristics:** – Very good if **ExposureMode** = ***Manual*** – Adaptively reduced if **ExposureValue** < 13200 µs or **ExposureMode** = ***External*** • **Usage:** Best suited for high-dynamic range (HDR) light conditions
On_Fast	Acquire and readout image at same time: • **NIR sensitivity:** Increased NIR sensitivity during total exposure time • **Anti-blooming characteristics:** Reduced anti-blooming characteristics • **Usage:** Best suited for low-light applications and small exposure times

Offset

OffsetValue – Integer – R/W

Range: [0-255] Default: *0*

Brightness (aka black level). Setting ***GainValue*** does not change the ***Offset-Value***.

SubstrateVoltage

VsubValue – Uint32 – R/C

Range: [Camera dependent] Units: mV

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

Whitebalance

Unlike Hue or ***ColorTransformationControl***, 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] Default: *5* 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] Default: *100* Units: percent

Determines how fast the auto white balance algorithm updates.

WhitebalMode – Enum – R/W

Manual	[Default] 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	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: [Camera 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.

WhitebalValueBlue – Uint32 – R/W

Range: [Camera 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 and blue accordingly.

DeviceStatus

DeviceTemperatureMainboard – Float32 – R

Units: Degree Celsius Resolution: 0.031 Accuracy: ±1 °C

Camera internal temperature measured at the internal control board.

DeviceTemperatureSensor – Float32 – R

Units: Degree Celsius Resolution: 0.031 Accuracy: ±1 °C

Camera internal temperature measured at the sensor.

EventControl

Event controls allow the enabling of various camera events to be transmitted to the host computer, triggering a registered event callback function.

www

See *PvCameraEventCallbackRegister* in **AVT PvAPI Manual**:

http://www.alliedvision.com/fileadmin/content/documents/products/software/software/PvAPI/docu/PvAPI_SDK_Manual.pdf

EventID

EventAcquisitionStart – Uint32 – R/C	**40000**
EventAcquisitionEnd – Uint32 –R/C	**40001**
EventFrameTrigger – Uint32 – R/C	**40002**
EventFrameTriggerReady– Uint32 – R/C	**40018**
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**
EventFrameTriggerReady – Uint32 – R/C	**40018**
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

Default: *Off*. 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 [Default]	*AcquisitionEnd*
FrameTrigger	*FrameTriggerReady*
AcquisitionRecordTrigger	*ExposureEnd*
PtpSyncLocked	*PtpSyncLost*
SyncIn1Fall	*SyncIn1Rise*
SyncIn2Fall	*SyncIn2Rise*
SyncIn3Fall	*SyncIn3Rise*
SyncIn4Fall	*SyncIn4Rise*

EventsEnable1 – Uint32 – R/W

Default: *0*. Bit field of all events. Bits correspond to last two digits of ***EventID***. For example, *Bit 1* is ***EventAcquisitionStart***, *Bit 2* is ***EventAcquisitionEnd***, and *Bit 10* is ***EventSyncIn1Rise***. This is an alternative to setting each event individually using the ***EventNotification*** and ***EventSelector*** method.

GigE

BandwidthCtrlMode – Enum – R/W

Select the desired mode of bandwidth control.

StreamBytesPerSecond	[Default] See the ***StreamBytesPerSecond*** control for more information
SCPD	Stream channel packet delay expressed in time-stamp counter units. This mode may be used to limit the rate of data from the camera to the host. It works by inserting a delay between successive stream channel packets, e.g. the longer the delay, the slower the data rate. 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 Default: *FALSE*

Enables camera to send GigE Vision Standard Protocol chunk data with an image. Currently implemented chunk data:

[Bytes 1 – 4] Acquisition count

[Byte 5]
These 8 bits indicate the following EF lens settings:

- *Bit 7 (Error)*: When this bit is set to 1, the EF lens is in an error state, bits 2 – 5 indicate enumerated value of last error, and all other bits and Bytes will be 0.
- *Bit 6 (Lens attached)*: When this bit is set to 1, an EF lens is attached to camera.
- *Bit 5 (Auto focus)*: When this bit is set to 1, the EF lens manual/auto focus switch is set to the auto focus position.
- *Bits 2 – 4 (Last error)*: Enumerated error value:
  - 0: No error detected
  - 1: Lens failed query by camera
  - 2: Lens communication error (can occur when removing lens)
  - 3: Lens communication error (can occur when removing lens)
  - 4: Lens remained busy for longer than 10 seconds
  - 5: Lens focus “Zero Stop” not detected
  - 6: Lens focus “Infinity Stop” not detected
- *Bits 0 – 1*: Upper 2 bits of focus percentage value (see **Byte 6**).

[Byte 6]
These 8 bits in conjunction with bits 0 – 1 of Byte 5, indicate the current focus position of the EF lens in (percentage of maximum focus range) * 10 (i.e. 1000 = 100 percent = Infinity Stop).
If the lens manual/auto focus switch is in the manual position these bits will be 0.

[Byte7]These 8 bits indicate the current aperture position of the EF lens in Dn. To convert Dn to FStop value, use formula: FStop = 2 (Dn – 8) /16.
[Byte 8] These 8 bits indicate the current focal length of the EF lens in mm.
[Bytes 9 – 12] Exposure value in µs.
[Bytes 13 – 16] Gain value in dB. For GT1930L and GT1930LC cameras: Gain value in tenths of dB (i.e. 201 represents 20.1 dB)
[Bytes 17 – 18] Sync in levels. A bit field. Bit 0 is sync-in 0, bit 1 is sync-in 1, etc. A bit value of 1 = level high, and a bit value of 0 = level low.
[Bytes 19 – 20] Sync out levels. A bit field. Bit 0 is sync-out 0, bit 1 is sync-out 1, etc. A bit value of 1 = level high, and a bit value of 0 = level low.
[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. AncillaryButter.

Note Camera cannot be acquiring image data while modifying ***ChunkModeActive***.

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.

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.
The user can set the value but internally it is overwritten to 5 for PvAPI v1.26.

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] Default: *1%* 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.

GvspRetries – Uint32 – R/W

Range: [1 – 100] Default: *3*

Maximum number of resend requests that the host driver will attempt before marking a packet dropped.

GvspSocketBuffersCount – Enum – R/W

Possible values: 256, 512, 1024, 2048, 4096, 8192 Default: *512*

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] Default: *50* 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] Default: *2500* Units: ms

The driver sends a heartbeat request packet to the camera every ***HeatbeatInterval*** milliseconds. If the camera fails to respond to the heartbeat request (200ms timeout), a retry is sent 200 ms later. After ***GvcpRetries*** (5 for PvAPI v1.26) times with no response, a camera unplugged event is returned by the driver.

Note

- ***HeatbeatInterval*** may be modified, but is overwritten to ***HeartbeatTimeout*** - 2500 on PvCameraOpen / SampleViewer open. This ensures driver sends unplugged event, and camera closes stream and control channel at same time.
- PvAPI users: see *PvLinkCallbackRegister* to register a callback function on unplug event.

HeartbeatTimeout – Uint32 – R/W

Range: [500 – 3,600,000] Default: *6000* Units: ms

Timespan for which the camera waits for a heartbeat packet. 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 Default: *Off*

Enables multicast mode. In order to enable this, the camera must not be streaming.

MulticastIPAddress – String – R/W

Set the multicast IP address.

NonImagePayloadSize – Unit32 – R

Units: Bytes

Size of chunk mode data. If ***ChunkModeActive*** = *FALSE*, ***NonImagePayloadSize*** = *0*.

PacketSize – Uint32 – R/W

Range: [Camera dependent] Units: Bytes

Determines the Ethernet packet size. Generally, this number should be set to as large as the network adapter 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 adapter 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 adapter. A ***PacketSize*** of 1500 is a safe setting which all GigE 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.

PayloadSize – Unit32 – R

Units: Bytes

Total size of payload in bytes.

- If ***ChunkModeActive*** = *TURE*:
  PayloadSize = ***TotalBytesPerFrame*** + ***NonImagePayloadSize*** + ***8***
- If ***ChunkModeActive*** = *FALSE*:
  PayloadSize = ***TotalBytesPerFrame***

PTP

Precision Time Protocol (PTP) manages clock synchronization of multiple devices across an Ethernet network, with ±1 µs tolerance. Once the clocks of the devices are synchronized, a synchronous software trigger can be sent to AVT cameras via the ***PtpAcquisitionGateTime*** control. On AVT GigE cameras, the device clock is represented by the camera ***TimeStampValue*** attribute.

www

For more information on PTP, see the IEEE 1588-2008 standard:

http://standards.ieee.org/findstds/standard/1588-2008.html

PtpAcquisitionGateTimeHi – Uint32 – R/W

Range: [0 – (2^{32}-1)] Default: *0* Units: Camera clock ticks*2^{32}

Upper 32 bits of ***PtpAcquisitionGateTime***. Used to schedule a synchronized "software trigger" on multiple PTP synchronized devices. Must be set beyond current camera ***TimeStampValue***, i.e., ***TimeStampValue*** >= ***PtpAcquisitionGateTime***. When set below ***TimeStampValue***, image acquisition stalls. ***PtpAcquisitionGateTime*** resets to zero when ***PtpMode*** set to *Off*.

PtpAcquisitionGateTimeLo – Uint32 – R/W

Range: [0 – (2^{32}-1)] Default: *0* Units: Camera clock ticks

Lower 32 bits of ***PtpAcquisitionGateTime***. See ***PtpAcquisitionGateTimeHi***.

PtpMode – Enum – R/W

Controls the PTP device behavior.

Note

If using the camera event channel, a ***EventPtpSyncLost*** is sent if ***PtpMode*** is changed. ***EventPtpSyncLocked*** is sent once PTP synchronization is reestablished.

Off	[Default] This device's ***TimeStampValue*** is not synchronized with any other device. ***PtpAcquisitionGateTime*** resets to zero
Slave	This device's ***TimeStampValue*** is altered to align with a master device's clock

Master	This device's ***TimeStampValue*** is the master clock. All other PTP enabled slave devices synchronize their clock to this camera
Auto	This device uses the IEEE1588 best master clock algorithm to determine which device is master, and which are slaves. It may be assigned as either. There may be several state transitions prior to synchronization

PtpStatus – Enum – R

State of the PTP operation.

Disabled	[Default] Device ***PtpMode*** is set to *Off*
Initializing	PTP is being initialized. If one camera / PTP device is being initialized, all devices statuses are set to initializing. This state appears very briefly
Listening	Device is listening for other PTP enabled devices. The purpose of this state is to determine which device will act as master
Master	Device acting as master clock. If a better master clock is determined, device will go to *Listening*, *Uncalibrated*, and finally *Slave*
Passive	If there are 2 or more devices with ***PtpMode*** = *Master,* this device has an inferior clock and is not synchronized to the master
Uncalibrated	PTP synchronization not yet achieved. Slave(s) are synching with master
Slave	PTP synchronization between this device and master is achieved. Device is acting as a slave to another device's master clock

Note

PTP capable cameras with firmware < 1.54.11026 have ***PtpStatus*** = [*Off*, *Master*, *Synching*, *Slave*, *Error*].

StreamBytesPerSecond – Uint32 – R/W

Range: [1,000,000 – 124,000,000 (248,000,000 for GX in LAG mode)]

Units: Bytes/s

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:

StreamBytesPerSecond = 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.

Note

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 Default: *TRUE*

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*, as these mode are not bandwidth limited.

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

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

Control on-camera image storage; this control is like a “pause” button for the image stream.

On	Images remain stored on the camera, and are not transmitted to the host
Off	[Default] The image stream resumes, and any stored images are sent to the host

Timestamp

TimeStampFrequency – Uint32 – R/C

Units: Hz

Camera clock frequency. Timebase for ***TimeStampValue***.

Note

PvAPI users: 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. Not possible while PTP enabled (***PtpMode*** = *Master,* or *Auto*).

TimeStampValueHi – Uint32 – R

Default: *0* 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

Default: *0* 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 - Camera dependent] Units: rows

The vertical size of the ROI rectangle.

RegionX – Uint32 – R/W

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

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

ImageMode

BinningHorizontalMode – Enum – R/W

Determines whether the result of binned pixels is averaged or summed up. Changing ***BinningHorizontalMode*** also changes ***BinningVerticalMode***.

Sum	[Default] Binning is accomplished by summing the charge / gray value of adjacent pixels on sensor
Average	Binning is accomplished by averaging the charge / gray value of adjacent pixels on sensor. This increases SNR by SQRT(number of binned pixels)

BinningVerticalMode – Enum – R/W

Determines whether the result of binned pixels is averaged or summed up. Changing ***BinningVerticalMode*** also changes ***BinningHorizontalMode*** .

Sum	[Default] Binning is accomplished by summing the charge / gray value of adjacent pixels on sensor
Average	Binning is accomplished by averaging the charge / gray value of adjacent pixels on sensor. This increases SNR by SQRT(number of binned pixels)

BinningX – Uint32 – R/W

Range: [1 – Camera dependent] Default: *1*

The horizontal binning factor. Binning is the summing of charge (for CCD sensors) or gray value (for CMOS sensors) of adjacent pixels on a sensor, giving a lower resolution image, but at full region of interest. Image sensitivity is also improved due to summed pixel charge / gray value.

Note

- ***BinningX*** and ***DecimationHorizontal*** are mutually exclusive. Setting ***BinningX*** > 1 forces ***DecimationHorizontal*** to 1.
- **Color cameras only**: Color information is lost while binning is active due to summing of adjacent different filtered pixels on the Bayer filter array.

BinningY – Uint32 – R/W

Range: [1 – Camera dependent] Default: *1*

The vertical binning factor. Binning is the summing of charge (for CCD sensors) or gray value (for CMOS sensors) of adjacent pixels on a sensor, giving a lower resolution image, but at full region of interest. Image sensitivity is also improved due to summed pixel charge / gray value.

Note

- ***BinningY*** and ***DecimationVertical*** are mutually exclusive. Setting ***BinningY*** > 1 forces ***DecimationVertical*** to 1.
- **Color cameras only**: Color information is lost while binning is active due to summing of adjacent different filtered pixels on the Bayer filter array.

DecimationHorizontal – Integer – R/W

Range: [1–8] Default: *1*

Decimation (also known as sub-sampling) is the process of skipping neighboring pixels (with the same color) while being read out from the CCD chip. ***DecimationHorizontal*** controls the horizontal sub-sampling of the image. There is no increase in the frame rate with horizontal sub-sampling.

1	Off
2	2x reduction factor. 2 of 4 columns displayed
4	4x reduction factor. 2 of 8 columns displayed
8	8x reduction factor. 2 of 16 columns displayed

Note

- Writing an invalid number for ***DecimationHorizontal*** will round up to next valid mode. For example, 5 rounds up to 8.
- ***DecimationHorizontal*** and ***BinningX*** are mutually exclusive. Setting ***DecimationHorizontal*** > 1 forces ***BinningX*** to 1.

DecimationVertical – Integer – R/W

Range: [1–8] Default: *1*

Decimation (also known as sub-sampling) is the process of skipping neighboring pixels (with the same color) while being read out from the CCD chip. ***DecimationVertical*** controls the vertical sub-sampling of the image. There is increase in frame rate with vertical sub-sampling.

1	Off
2	2x reduction factor. 2 of 4 rows displayed
4	4x reduction factor. 2 of 8 rows displayed
8	8x reduction factor. 2 of 16 rows displayed

Note

- Writing an invalid number for ***DecimationVertical*** will round up to next valid mode. For example, 5 rounds up to 8.
- ***DecimationVertical*** and ***BinningY*** are mutually exclusive. Setting ***DecimationVertical*** > 1 forces ***BinningY*** to 1.

Note Writing an invalid number for ***DecimationVertical*** will round up to next valid mode. For example, 5 rounds up to 8.

www For more information on the decimation process, see:

http://www.alliedvision.com/fileadmin/content/documents/products/cameras/various/appnote/Decimation.pdf

ReverseX – Boolean – R/W

Possible values: True, False Default: *False*

Flips the image sent by device horizontally. The region of interest (ROI) is applied after flipping.

ReverseY – Boolean – R/W

Possible values: True, False Default: *False*

Flips the image sent by device vertically. The region of interest (ROI) is applied after flipping.

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.

DeviceScanType – Enum – R/C

Scan type of the camera, *Areascan*.

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. Firmware 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 dependent).

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 is camera model dependent.

StatusLed1

Indicates status of LED1.

StatusLedInvert - Enum - R/W

Possible values: On, Off

Polarity applied to the status LED.

Note

On: yellow LED

Off: green LED

StatusLed1Mode - Enum - R/W

Determines the behavior of the ***StatusLed1***.

GPO	Configured to be a general purpose output, control of which is assigned to ***StatusLedGpoLevels***
AcquisitionTriggerReady	Active once the camera has been recognized by the host PC and is ready to start acquisition
FrameTriggerReady	Becomes active when the camera is in a state that will accept the next frame trigger
FrameTrigger	This is the logic trigger signal inside of the camera. It is initiated by an external trigger or software trigger
Exposing	[Default] Exposure in progress
FrameReadout	Becomes active at the start of frame readout
Imaging	Exposing or frame readout. Active when the camera is exposing or reading out frame data
Acquiring	Becomes active at the start of acquisition
SyncIn1/2/3/4	External input *SyncIn1/2/3/4*
Strobe1	Source is strobe timing unit
CCDTemperatureOK	Only for cameras that support this feature: indicates if camera has reached the desired temperature value

StatusLedGpoLevels - Enum - R/W

Possible values: *RegStatusLedLevels*

Status LED levels in GPO mode.

Note

StatusLedInvert can invert these values.

Strobe

1

Strobe is an internal signal generator for on-camera clocking functions. 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.

Strobe1ControlledDuration – Enum – R/W

Possible values: On, Off Default: *Off*

When enabled, the ***Strobe1Duration*** control is valid.

Strobe1Delay – Uint32 – R/W

Range: [0 - Camera dependent] Default: *0* Units: µs

Delay of start of strobe signal.

Strobe1Duration – Uint32 – R/W

Range: [0 - Camera dependent] Default: *0* 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	[Default] 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*
SyncIn3	Active when there is an external trigger at *SyncIn3*
SyncIn4	Active when there is an external trigger at *SyncIn4*

Note

For detailed information see the camera waveform diagrams provided in the camera manuals.

SyncIn1

SyncIn1GlitchFilter – Uint32 – R/W

Range: [0 – 50000] Default: *0* Units: relative

Ignores glitches on the *SyncIn1* input line with pulse duration less than set value. Units are approximately accurate to nanoseconds. Exact units are camera model and input dependent.

Note Setting **this** value increases latency of ***FrameTrigger*** by same amount.

SyncIn2/3/4

Analogous to *SyncIn1*.

SyncInLevels – Uint32 – R

A bit field, 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

Possible values: On, Off Default: *Off*

When enabled, reverses the polarity of the signal output by *SyncOut1*.

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
Exposing	Active for the duration of sensor exposure
FrameReadout	Active during frame readout, i.e. the transferring of image data from the CCD to camera memory
Imaging	Active when the camera is exposing or reading out frame data
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*
SyncIn3	Active when there is an external trigger at *SyncIn3*

SyncIn4	Active when there is an external trigger at *SyncIn4*
Strobe1	The output signal is controlled according to Strobe1 settings
CCDTemperatureOK	Only for cameras that support this feature: indicates if camera has reached the desired temperature value

Note

For detailed information see the camera waveform diagrams provided in the camera manuals.

SyncOut2/3/4

Analogous to *SyncOut1*.

SyncOutGpoLevels – Uint32 – R/W

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

Stats

CCDTemperatureOK – Uint32 – R

Momentary temperature status of the CCD sensor. Indicates if CCD sensor has desired cooling temperature.

0	The CCD sensor is too hot. Acquired image data may have higher noise than expected or contain erroneous pixels at long exposure times
1	The CCD sensor temperature is in the desired temperature range. Acquired image data are OK

StatDriverType – Enum – R

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

Frame rate of the camera.

StatFramesCompleted – Uint32 – R

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

Note

PvAPI programmers: 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

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

Note

PvAPI programmers: 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

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

StatPacketsMissed – Uint32 – R

The number of packets missed since the start of imaging.

StatPacketsReceived – Uint32 – R

The number of packets received since the start of imaging.

StatPacketsRequested – Uint32 – R

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.

StatPacketsResent – Uint32 – R

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

Index

L

M

N

O

P

R

S

For technical support, please contact support@alliedvision.com.
For comments or suggestions regarding this document, please contact info@alliedvision.com.

Disclaimer

Due to continual product development, technical specifications may be subject to change without notice. All trademarks are acknowledged as property of their respective owners. We are convinced that this information is correct. We acknowledge that it may not be comprehensive. Nevertheless, AVT cannot be held responsible for any damage in equipment or subsequent loss of data or whatsoever in consequence of this document.