

# AVT GigE Vision Cameras



## Camera and Driver Attributes

**Bigeye G Firmware v3.1.44.6**

**Mako G Firmware v1.52**

**Manta Firmware v1.52**

**Prosilica Firmware v1.52**

V1.1.1

06 September 2013

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

 **ALLIED**  
Vision Technologies

# Contents

<b>Contacting Allied Vision Technologies</b>	5
<b>Introduction</b>	6
Important notes	6
Document history	7
Symbols used in this manual	8
Additional information	9
<b>Camera attributes</b>	10
<b>Acquisition</b>	10
Trigger	10
AcquisitionAbort – Command	13
AcquisitionFrameCount – Uint32 – R/W	13
AcquisitionMode – Enum – R/W	13
AcquisitionStart – Command	13
AcquisitionStop – Command	13
RecorderPreEventCount – Uint32 – R/W	13
<b>ConfigFile</b>	14
ConfigFileIndex – Enum – R/W	14
ConfigFileLoad – Command	14
ConfigFilePowerup – Enum – R/W	14
ConfigFileSave – Command	14
<b>Controls</b>	14
ColorTransformationControl	14
DSP	16
DefectMask	16
EdgeFilter – Enum – R/W	17
EFLensControl	17
Exposure	19
Gain	22
Gamma – Float32 – R/W	23
Hue – Float32 – R/W	23
IODMode - Enum - R/W	23
LensDrive	24
Iris	24
Saturation – Float32 – R/W	26
LUTControl	26
NirMode – Enum – R/W	28
Offset	28
SubstrateVoltage	29
Whitebalance	29
<b>DeviceStatus</b>	30
DeviceTemperatureMainboard – Float32 – R	30
DeviceTemperatureSensor – Float32 – R	30

<b>EventControl</b> .....	30
EventID.....	30
EventNotification – Enum – R/W .....	31
EventSelector – Enum – R/W .....	31
EventsEnable1 – Uint32 – R/W.....	31
<b>GigE</b> .....	32
BandwidthCtrlMode – Enum – R/W .....	32
ChunkModeActive – Boolean – R/W.....	32
NonImagePayloadSize – Unit32 – R .....	32
PayloadSize – Unit32 – R .....	32
StreamBytesPerSecond – Uint32 – R/W .....	33
StreamFrameRateConstrain – Boolean – R/W.....	33
StreamHold .....	33
Timestamp.....	34
<b>ImageFormat</b> .....	35
ROI .....	35
PixelFormat – Enum – R/W.....	35
ReverseX – Boolean – R/W .....	36
TotalBytesPerFrame – Uint32 – R .....	36
<b>ImageMode</b> .....	36
BinningX – Uint32 – R/W.....	36
BinningY – Uint32 – R/W .....	36
DecimationHorizontal – Integer – R/W .....	36
DecimationVertical – Integer – R/W .....	37
<b>Info</b> .....	37
CameraName – String – R/W.....	37
DeviceFirmwareVersion – String – R/C.....	37
DevicemodelName – String – R/W.....	37
DevicePartNumber – String – R/C .....	37
DeviceScanType – Enum – R/C .....	37
DeviceSerialNumber – String – R/C .....	37
DeviceVendorName – String – R/C .....	38
Firmware.....	38
Part.....	38
Sensor.....	38
UniqueID – Uint32 – R/C .....	38
<b>IO</b> .....	39
StatusLed1 .....	39
StatusLedGpoLevels - Enum - R/W .....	39
Strobe .....	40
SyncIn1 .....	41
SyncIn2/3/4 .....	41
SyncInLevels – Uint32 – R.....	41
SyncOut1 .....	41
SyncOut2/3/4 .....	42
SyncOutGpoLevels – Uint32 – R/W.....	42
<b>Driver attributes</b> .....	43

<b>GigE</b> .....	43
Ethernet .....	43
IP .....	43
GvcpRetries – UInt32 – R/W .....	43
Gvsp .....	43
HeartbeatInterval – UInt32 – R/W .....	44
HeartbeatTimeout – UInt32 – R/W .....	44
Multicast .....	44
PacketSize – UInt32 – R/W .....	45
PTP .....	45
<b>Stats</b> .....	46
CCDTemperatureOK – UInt32 – R .....	46
StatDriverType – Enum – R .....	46
StatFilterVersion – String – R/C .....	46
StatFrameRate – Float32 – R .....	46
StatFramesCompleted – UInt32 – R .....	47
StatFramesDropped – UInt32 – R .....	47
StatPacketsErroneous – UInt32 – R .....	47
StatPacketsMissed – UInt32 – R .....	47
StatPacketsReceived – UInt32 – R .....	47
StatPacketsRequested – UInt32 – R .....	47
StatPacketsResent – UInt32 – R .....	47
<b>Disclaimer</b> .....	48

# Contacting Allied Vision Technologies

## Info



- **Technical information:**  
<http://www.alliedvisiontec.com>
- **Support:**  
[support@alliedvisiontec.com](mailto:support@alliedvisiontec.com)

### **Allied Vision Technologies GmbH (Headquarters)**

Taschenweg 2a  
07646 Stadtdroda, Germany  
Tel: +49 36428-677-0  
Fax: +49 36428-677-28  
e-mail: [info@alliedvisiontec.com](mailto: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](mailto: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](mailto:info@alliedvisiontec.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@alliedvisiontec.com](mailto:info@alliedvisiontec.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@alliedvisiontec.com](mailto:info@alliedvisiontec.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.alliedvisiontec.com/emea/products/cameras/gigabit-ethernet/manta.html>

## 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, SyncIn1Glitch-Filter 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

Table 1: Document History

V1.1.0	2013-Jul-05	<ul style="list-style-type: none"> <li>• Added Bigeye G camera controls</li> <li>• Added Mako G controls</li> <li>• Added contact information for Allied Vision Technologies (Shanghai) Co. Ltd.</li> <li>• Changed user access from R/V to R</li> <li>• Updated the following controls:           <ul style="list-style-type: none"> <li>– PayloadSize</li> <li>– EdgeFilter</li> <li>– Gamma</li> <li>– Hue</li> <li>– IrisVideoLevelMax</li> <li>– IrisVideoLevelMin</li> <li>– Saturation</li> <li>– LUTControl</li> <li>– BandwidthCtrlMode</li> <li>– StreamHoldEnable</li> </ul> </li> </ul>
V1.1.1	2013-Sep-06	<ul style="list-style-type: none"> <li>• Added the EF lens controls</li> <li>• Added ReverseX control on page 36</li> <li>• Updated the DefectMaskPixelEnable, Eventcontrol, and DeviceStatus controls</li> </ul>

Table 1: Document History

## Symbols used in this manual

<u>Note</u>	This symbol highlights important information.
	
<u>www</u>	This symbol highlights URLs for further information. The URL itself is shown in blue.  Example: <a href="http://www.alliedvisiontec.com">http://www.alliedvisiontec.com</a>

## 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.alliedvisiontec.com/emea/support/downloads/software.html>

### Third-party software

In addition to the software provided by AVT, there are numerous GigE Vision Standard compliant third-party software options available. In general, third-party software provides increased functionality such as image processing and video recording.

**www**

For a list of compliant third-party software see:



<http://www.alliedvisiontec.com/emea/products/software/third-party-software.html>

# Camera attributes

Camera attributes affect settings on the camera itself, whereas, driver attributes affect the host PvAPI driver.

## Acquisition

### Trigger

#### AcqEnd

##### AcqEndTriggerEvent – Enum – R/W

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

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

##### AcqEndTriggerMode – Enum – R/W

Determines if end of acquisition initiated by an external hardware trigger.

<i>SyncIn1</i>	Trigger at <i>SyncIn1</i> to be associated with this control
<i>SyncIn2</i>	Trigger at <i>SyncIn2</i> to be associated with this control
<i>SyncIn3</i>	Trigger at <i>SyncIn3</i> to be associated with this control
<i>SyncIn4</i>	Trigger at <i>SyncIn4</i> to be associated with this control
<i>Disabled</i>	[Default] No external trigger. Acquisition must be stopped with the <b>AcquisitionStop</b> 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.

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

### **AcqRecTriggerMode – Enum – R/W**

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

<i>SyncIn1</i>	[Default] Trigger at <i>SyncIn1</i> to be associated with this control
<i>SyncIn2</i>	Trigger at <i>SyncIn2</i> to be associated with this control
<i>SyncIn3</i>	Trigger at <i>SyncIn3</i> to be associated with this control
<i>SyncIn4</i>	Trigger at <i>SyncIn4</i> to be associated with this control
<i>Disabled</i>	No external trigger. Unlike <b>AcqStart</b> and <b>AcqEnd</b> , 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.

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

### **AcqStartTriggerMode – Enum – R/W**

Determines if start of acquisition initiated by an external hardware trigger.

<i>SyncIn1</i>	Trigger at <i>SyncIn1</i> to be associated with this control
<i>SyncIn2</i>	Trigger at <i>SyncIn2</i> to be associated with this control
<i>SyncIn3</i>	Trigger at <i>SyncIn3</i> to be associated with this control
<i>SyncIn4</i>	Trigger at <i>SyncIn4</i> to be associated with this control
<i>Disabled</i>	[Default] No external trigger. Acquisition must be started with the <b>AcquisitionStart</b> API command

### **FrameRate – Float32 – R/W**

Range: [Camera dependent] Units: Frames per second

When **FrameStartTriggerMode** is set to *FixedRate*, this control specifies the frame rate.

### **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:  $\mu\text{s}$

Start of frame is delayed **FrameStartTriggerDelay**  $\mu\text{s}$  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.

<i>EdgeRising</i>	[Default] Rising edge trigger
<i>EdgeFalling</i>	Falling edge trigger
<i>EdgeAny</i>	Rising or falling edge
<i>LevelHigh</i>	Active high signal
<i>LevelLow</i>	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.

<i>Freerun</i>	[Default] Frame triggers generated on-camera, at maximum supported frame rate depending on the exposure time and region of interest size
<i>SyncIn1</i>	External trigger <i>SyncIn1</i>
<i>SyncIn2</i>	External trigger <i>SyncIn2</i>
<i>SyncIn3</i>	External trigger <i>SyncIn3</i>
<i>SyncIn4</i>	External trigger <i>SyncIn4</i>
<i>FixedRate</i>	Frame triggers generated on-camera, at frame rate defined by <i>FrameRate</i> attribute
<i>Software</i>	Software initiated frame trigger. See <b>FrameStartTrigger-Software</b> command

### **FrameStartTriggerOverlap – Enum – R/W**

Does not work with Software triggering. Only external.

<i>Off</i>	[Default] When <i>Off</i> , any external trigger received before <b>FrameTriggerReady</b> signal is high is ignored
<i>PreviousFrame</i>	When <i>PreviousFrame</i> , any external trigger received before <b>FrameTriggerReady</b> 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.

<i>Continuous</i>	[Default] The camera will continuously receive frame triggers
<i>SingleFrame</i>	The camera will only receive a single frame trigger event. Further frame triggers will be ignored until acquisition is stopped and restarted
<i>MultiFrame</i>	The camera will receive <b>AcquisitionFrameCount</b> number of frame triggers. Further frame triggers will be ignored until acquisition is stopped and restarted
<i>Recorder</i>	<p>The camera will continuously capture images into camera memory, but will not send them to the host until an <b>AcqRec</b> trigger signal is received. Further, <b>AcqRec</b> trigger events will be ignored until acquisition is stopped and restarted.</p> <p>This feature allows returning <b>RecorderPreEventCount</b> number of frames before the trigger event, and <b>AcquisitionFrameCount</b> minus <b>RecorderPreEventCount</b> frames after the trigger.</p> <p>When <b>AcqRec</b> 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</p>

## 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 **AcquisitionFrameCount** 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.

## 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.  
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**

<i>Off</i>	No color transformation
<i>Manual</i>	Manually set <b>ColorTransformationValue</b> matrix coefficients
<i>Temp6500K</i>	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.

**DefectMask**

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

www

See the AVT modular concept document, or contact your AVT sales representative for more information:


<http://www.alliedvisiontec.com/us/support/downloads/product-literature/avt-modular-concept.html>
**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.

<i>Enabled</i>	[Default] Enables masking of defective columns
----------------	--

<i>Disabled</i>	Disables masking of defective columns
-----------------	---------------------------------------

**DefectMaskPixelEnable – Enum – R/W**

Currently NOT implemented.

## EdgeFilter – Enum – R/W

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

<i>Smooth2</i>	Most blur
<i>Smooth1</i>	Slight blur
<i>Off</i>	No blur or sharpness applied
<i>Sharpen1</i>	Slight sharp
<i>Sharpen2</i>	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 AVT GigE cameras.

### 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.

<i>AutoFocus</i>	Switch is in auto focus (AF) position
<i>ManualFocus</i>	Switch is in manual focus (MF) position

## **EFLensFStop**

### **EFLensFStopCurrent – Float32 – R/W**

Range: [EFLensFStopMin – EFLensFStopMax] Unit: F-Stop

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 – Float32 – R/W**

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 – Float32 – R/W**

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**

Range: [1 – 8] Units: F-Stop/8

Size of increments/decrements in *EFLensFStopCurrent* when using *EFLensFStopIncrease* and *EFLensFStopDecrease* commands, respectively.

## **EFLensStatus**

### **EFLensID – Integer – R**

Identification value of the attached EF lens.

### **EFLensLastError – Enum – R**

Most recently detected error.

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

### **EFLensState – Enum – R**

Current EF lens state.

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

<i>Mean</i>	[Default] The arithmetic mean of the histogram of the current image is compared to <i>ExposureAutoTarget</i> , and the next image adjusted in exposure time to meet this target. Bright areas are allowed to saturate
<i>FitRange</i>	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:  $\mu$ s

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  $1 \times 10^6$  / (desired frame rate).

### **ExposureAutoMin – UInt32 – R/W**

Range: [Camera dependent] Default: *Camera dependent* Units:  $\mu$ s

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%

The total pixels from top of the distribution that are ignored by the auto exposure algorithm.

### **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: [1 – 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**

<i>Manual</i>	[Default] The camera exposure time is fixed by <b>ExposureValue</b> parameter
<i>Auto</i>	The exposure time will vary continuously according to the scene illumination. The <i>Auto</i> exposure function operates according to the Auto and DSP controls
<i>AutoOnce</i>	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 <i>AutoOnce</i> function operates according to the Auto and DSP controls
<i>External</i>	When <b>ExposureMode</b> is set to <i>External</i> the exposure time will be controlled by an external signal appearing on <i>SyncIn1</i> or <i>SyncIn2</i> . In order for this feature to work, the parameter <b>Frame-StartTriggerMode</b> must be set to <i>SyncIn1</i> or <i>SyncIn2</i>

### **ExposureValue – UInt32 – R/W**

Range: [Camera dependent]      Units:  $\mu\text{s}$   
 The sensor integration time.

### **Shutter - Enum - R/W**

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

<i>Off</i>	Deactivate the mechanical shutter. Use this mode, if you operate the camera with pulsed light sources
<i>On</i>	[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
<i>SyncIn1</i>	Enables or disables the mechanical shutter dependent on the level of <i>SyncIn1</i>
<i>SyncIn2</i>	Enables or disables the mechanical shutter dependent on the level of <i>SyncIn2</i>
<i>SyncIn3</i>	Enables or disables the mechanical shutter dependent on the level of <i>SyncIn3</i>
<i>SyncIn4</i>	Enables or disables the mechanical shutter dependent on the level of <i>SyncIn4</i>
<i>SyncIn5</i>	Enables or disables the mechanical shutter dependent on the level of <i>SyncIn5</i>

#### **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.

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.

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



#### **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: dB

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

#### **GainAutoMin – UInt32 – R/W**

Range: [0 – Camera dependent] Default: 0 Units: dB

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%

The total pixels from top of the distribution that are ignored by the auto gain algorithm.

#### **GainAutoRate – UInt32 – R/W**

Range: [0 – 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**

<i>Manual</i>	[Default] The camera gain is fixed by <b>GainValue</b> parameter
<i>Auto</i>	The gain will vary continuously according to the scene illumination. The <i>Auto</i> function operates according to the Auto and DSP controls
<i>AutoOnce</i>	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 <i>AutoOnce</i> function operates according to the Auto and DSP controls
<i>External</i>	When <b>ExposureMode</b> is set to External the exposure time will be controlled by an external signal appearing on <i>SyncIn1</i> or <i>SyncIn2</i> . In order for this feature to work, the parameter <b>Frame-StartTriggerMode</b> must be set to <i>SyncIn1</i> or <i>SyncIn2</i>

### **GainValue – UInt32 – R/W**

Range: [Camera dependent]    Default: 0    Units: dB

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

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 = (Input)<sup>Gamma</sup>  
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.

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

**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.

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

### LensDriveDuration – UInt32 – R/W

Range: [0 – 5000] Units:  $\mu$ s

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.

<i>Disabled</i>	[Default] Disable auto-iris
<i>Video</i>	Enable video auto iris. Video-type lenses only
<i>VideoOpen</i>	Fully open the iris. Video-type lenses only
<i>VideoClosed</i>	Full close the iris. Video-type lenses only
<i>P IrisAuto</i>	Enable P auto iris. P-Iris lenses only
<i>P IrisManual</i>	Manually control iris via <b>LensP IrisPosition</b> attribute P-Iris lenses only
<i>DC Iris</i>	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 <b>IrisAutoTarget</b>
DC-iris lenses	[0-100]	Attempts to match <b>IrisAutoTarget</b>

**IrisVideoLevelMax – UInt32 – R/W**

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

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.alliedvisiontec.com/fileadmin/content/PDF/Support/Application\\_Notes/AppNote\\_-\\_P-iris\\_Lenses\\_Supported\\_by\\_Prosilica\\_GT\\_Cameras.pdf](http://www.alliedvisiontec.com/fileadmin/content/PDF/Support/Application_Notes/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 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. Applied post-bayer interpolation. Only available on color **PixelFormat**s 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 information depending on **LUTSelector**.

**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 look-up table 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:  $[0 - (2^{\text{LUTBitDepthIn}} - 1)]$       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 (depending on **LUTSelector**) will be applied.

<i>Luminance</i>	[Default] LUT is applied on all pixels
<i>Red</i>	LUT is applied on red pixels only
<i>Green</i>	LUT is applied on green pixels only
<i>Blue</i>	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 modes.

**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<sup>LUTBitDepthOut</sup> - 1)] Default: 4095

The value at entry **LUTIndex** of the LUT selected by **LUTSelector**.

**NirMode - Enum - R/W****Manta NIR models only.**

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

<i>Off</i>	NirMode set off. Acquire and readout image at same time: <ul style="list-style-type: none"> <li><b>NIR sensitivity:</b> No increased sensitivity in NIR range</li> <li><b>Anti-blooming characteristics:</b> As specified by sensor manufacturer</li> <li><b>Usage:</b> Best suited if you need very long exposure time</li> </ul>
<i>On_HighQuality</i>	[Default] Can't acquire and readout image at same time. The exposure time will always influence frame rate directly: <ul style="list-style-type: none"> <li><b>NIR sensitivity:</b> Increased NIR sensitivity, except for a very small portion of the exposure time, which is: <math>t_{NormalQE} = \text{MIN}(4300 \mu\text{s}, \text{ExposureValue}/4)</math></li> <li><b>Anti-blooming characteristics:</b> <ul style="list-style-type: none"> <li>– Very good if <b>ExposureMode = Manual</b></li> <li>– Adaptively reduced if <b>ExposureValue &lt; 13200 μs</b> or <b>ExposureMode = External</b></li> </ul> </li> <li><b>Usage:</b> Best suited for high-dynamic range (HDR) light conditions</li> </ul>
<i>On_Fast</i>	Acquire and readout image at same time: <ul style="list-style-type: none"> <li><b>NIR sensitivity:</b> Increased NIR sensitivity during total exposure time</li> <li><b>Anti-blooming characteristics:</b> Reduced anti-blooming characteristics</li> <li><b>Usage:</b> Best suited for low-light applications and small exposure times</li> </ul>

**Offset****OffsetValue - Integer - R/W**

Range: [0-255] Default: 0

Brightness (aka black level). Setting **GainValue** does not change the **OffsetValue**.

## 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**

<i>Manual</i>	[Default] Auto white balance is off. White balance can be adjusted directly by changing the <b>WhitebalValueRed</b> and <b>WhitebalValueBlue</b> parameters
<i>Auto</i>	White balance will continuously adjust according to the current scene. The <i>Auto</i> function operates according to the Auto and DSP controls
<i>AutoOnce</i>	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 <i>AutoOnce</i> 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:  $\pm 1^{\circ}\text{C}$

Camera internal temperature measured at the internal control board.

### **DeviceTemperatureSensor – Float32 – R**

Units: Degree Celsius    Resolution: 0.031    Accuracy:  $\pm 1^{\circ}\text{C}$

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.alliedvisiontec.com/fileadmin/content/PDF/Software/Prosilica\\_software/Prosilica\\_software\\_doc/PvAPI\\_SDK\\_Manual.pdf](http://www.alliedvisiontec.com/fileadmin/content/PDF/Software/Prosilica_software/Prosilica_software_doc/PvAPI_SDK_Manual.pdf)

## EventID

<b>EventAcquisitionStart – UInt32 – R/C</b>	<b>40000</b>
<b>EventAcquisitionEnd – UInt32 – R/C</b>	<b>40001</b>
<b>EventFrameTrigger – UInt32 – R/C</b>	<b>40002</b>
<b>EventFrameTriggerReady – UInt32 – R/C</b>	<b>40018</b>
<b>EventExposureEnd – UInt32 – R/C</b>	<b>40003</b>
<b>EventAcquisitionRecordTrigger – UInt32 – R/C</b>	<b>40004</b>
<b>EventPtpSyncLost – UInt32 – R/C</b>	<b>40005</b>
<b>EventPtpSyncLocked – UInt32 – R/C</b>	<b>40006</b>
<b>EventSyncIn1Rise – UInt32 – R/C</b>	<b>40010</b>
<b>EventSyncIn1Fall – UInt32 – R/C</b>	<b>40011</b>
<b>EventSyncIn2Rise – UInt32 – R/C</b>	<b>40012</b>
<b>EventSyncIn2Fall – UInt32 – R/C</b>	<b>40013</b>

<b>EventSyncIn3Rise – UInt32 – R/C</b>	<b>40014</b>
<b>EventSyncIn3Fall – UInt32 – R/C</b>	<b>40015</b>
<b>EventSyncIn4Rise – UInt32 – R/C</b>	<b>40016</b>
<b>EventSyncIn4Fall – UInt32 – R/C</b>	<b>40017</b>
<b>EventOverflow – UInt32 – R/C</b>	<b>65534</b>
Always on. Cannot be turned off with <b>EventSelector</b> or <b>EventsEnable1</b> . 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.	
<b>EventError – UInt32 – R/C</b>	<b>65535</b>
Always on. Cannot be turned off with <b>EventSelector</b> or <b>EventsEnable1</b> . 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:

<i>AcquisitionStart</i> [Default]	<i>AcquisitionEnd</i>
<i>FrameTrigger</i>	<i>FrameTriggerReady</i>
<i>AcquisitionRecordTrigger</i>	<i>ExposureEnd</i>
<i>PtpSyncLocked</i>	<i>PtpSyncLost</i>
<i>SyncIn1Fall</i>	<i>SyncIn1Rise</i>
<i>SyncIn2Fall</i>	<i>SyncIn2Rise</i>
<i>SyncIn3Fall</i>	<i>SyncIn3Rise</i>
<i>SyncIn4Fall</i>	<i>SyncIn4Rise</i>

## 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.

<i>StreamBytesPerSecond</i>	[Default] See the <b>StreamBytesPerSecond</b> control for more information
<i>SCPD</i>	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
<i>Both</i>	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	[Bytes 25 – 28]	reserved. 0
[Bytes 5 – 8]	reserved. 0	[Bytes 29 – 32]	reserved. 0
[Bytes 9 – 12]	exposure value.	[Bytes 33 – 36]	reserved. 0
[Bytes 13 – 16]	gain value	[Bytes 37 – 40]	reserved. 0
[Bytes 17 – 18]	sync in levels	[Bytes 41 – 44]	chunk ID. 1000
[Bytes 19 – 20]	sync out levels	[Bytes 45 – 48]	chunk length
Bytes 21 – 24]	reserved. 0		

PvAPI users see tPvFrame. AncillaryBuffer.

## **NonImagePayloadSize – Unit32 – R**

Units: Bytes

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

## **PayloadSize – Unit32 – R**

Units: Bytes

Total size of payload in bytes.

- If **ChunkModeActive** = TRUE:  
 $\text{PayloadSize} = \text{TotalBytesPerFrame} + \text{NonImagePayloadSize} + 8$
- If **ChunkModeActive** = FALSE:  
 $\text{PayloadSize} = \text{TotalBytesPerFrame}$

## **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.

<i>On</i>	Images remain stored on the camera, and are not transmitted to the host
<i>Off</i>	[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.

### **TimeStampValueHi – UInt32 – R**

Default: 0    Units: Camera clock ticks\* $2^{32}$

Time stamp, upper 32-bit.  $\text{TimeStampValueHi} * 2^{32} / \text{TimeStampFrequency}$  = 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.  $\text{TimeStampValueLo} / \text{TimeStampFrequency}$  = 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.

## 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.

## TotalBytesPerFrame – UInt32 – R

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: [Camera dependent]. The horizontal binning factor.

## BinningY – UInt32 – R/W

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

## DecimationHorizontal – Integer – R/W

Horizontal sub-sampling of the image. This reduces the horizontal resolution (width) of the image by the specified horizontal decimation factor. No increase in the frame rate.

1	[Default] Off
2	2 out of 4 decimation
3/4	2 out of 8 decimation
5/6/7/8	2 out of 16 decimation

## DecimationVertical – Integer – R/W

Vertical sub-sampling of the image. This reduces the vertical resolution (width) of the image by the specified vertical decimation factor. Frame rate is increased.

1	[Default] Off
2	2 out of 4 decimation
3/4	2 out of 8 decimation
5/6/7/8	2 out of 16 decimation

### Note



Each combination of vertical binning and horizontal decimation or vice versa is possible.

Combination of horizontal binning + horizontal decimation is not possible.

Combination of vertical binning + vertical decimation is not possible.

### www



See the application note for more information on the decimation process:

[http://www.alliedvisiontec.com/fileadmin/content/PDF/Support/Application\\_Notes/AppNote\\_-\\_Decimation.pdf](http://www.alliedvisiontec.com/fileadmin/content/PDF/Support/Application_Notes/AppNote_-_Decimation.pdf)

## 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 **Part-Number** 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*.

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

Delay of start of strobe signal.

### **Strobe1Duration – UInt32 – R/W**

Range: [0 - Camera dependent]    Default: 0    Units:  $\mu$ s

Duration of strobe signal.

### **Strobe1Mode – Enum – R/W**

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

<i>AcquisitionTriggerReady</i>	Active once the camera has been recognized by the host PC and is ready to start acquisition
<i>FrameTriggerReady</i>	Active when the camera is in a state that will accept the next frame trigger
<i>FrameTrigger</i>	[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
<i>Exposing</i>	Active for the duration of sensor exposure
<i>FrameReadout</i>	Active at during frame readout, i.e. the transferring of image data from the CCD to camera memory
<i>Imaging</i>	Active during exposure and readout
<i>Acquiring</i>	Active during an acquisition stream
<i>SyncIn1</i>	Active when there is an external trigger at <i>SyncIn1</i>
<i>SyncIn2</i>	Active when there is an external trigger at <i>SyncIn2</i>
<i>SyncIn3</i>	Active when there is an external trigger at <i>SyncIn3</i>
<i>SyncIn4</i>	Active when there is an external trigger at <i>SyncIn4</i>

**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*:

<i>GPO</i>	Configured to be a general purpose output, control of which is assigned to <i>SyncOutGpoLevels</i>
<i>AcquisitionTriggerReady</i>	Active once the camera has been recognized by the host PC and is ready to start acquisition
<i>FrameTriggerReady</i>	Active when the camera is in a state that will accept the next frame trigger
<i>FrameTrigger</i>	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

<i>Exposing</i>	Active for the duration of sensor exposure
<i>FrameReadout</i>	Active during frame readout, i.e. the transferring of image data from the CCD to camera memory
<i>Imaging</i>	Active when the camera is exposing or reading out frame data
<i>Acquiring</i>	Active during an acquisition stream
<i>SyncIn1</i>	Active when there is an external trigger at <i>SyncIn1</i>
<i>SyncIn2</i>	Active when there is an external trigger at <i>SyncIn2</i>
<i>SyncIn3</i>	Active when there is an external trigger at <i>SyncIn3</i>
<i>SyncIn4</i>	Active when there is an external trigger at <i>SyncIn4</i>
<i>Strobe1</i>	The output signal is controlled according to <i>Strobe1</i> settings
<i>CCDTemperatureOK</i>	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.

# Driver attributes

Driver controls affect the connection between the driver on the host side and the camera, leading to changes on both sides; whereas, camera controls have no influence on the driver. **Driver controls 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.

### 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 **GvspLookback-Window** 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: 3500 Units: ms

The driver sends heartbeat packets to the camera every **HeartbeatInterval** milliseconds.

**HeartbeatTimeout – UInt32 – R/W**

Range: [500 – 3,600,000] Default: 6000 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    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.

**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.

**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<sup>32</sup>-1)]    Default: 0    Units: Camera clock ticks\*2<sup>32</sup>

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<sup>32</sup>-1)]    Default: 0    Units: Camera clock ticks

Lower 32 bits of **PtpAcquisition** trigger time. See **PtpAcquisitionGateTimeHi**.

**PtpMode – Enum – R/W**

<i>Off</i>	[Default] This camera's <b>TimeStampValue</b> is not synchronized with any other device
<i>Master</i>	This camera's <b>TimeStampValue</b> is the master clock. All other PTP enabled slave devices synchronize their clock to this camera
<i>Slave</i>	This camera's <b>TimeStampValue</b> is altered to align with a master device's clock
<i>Auto</i>	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**

<i>Off</i>	[Default] Camera <b>PtpMode</b> set to Off
<i>Master</i>	This camera acting as master clock
<i>Synching</i>	PTP synchronization not yet achieved. Slave(s) are synching with master
<i>Slave</i>	PTP synchronization among devices achieved. This camera acting as a slave to another device's master clock

# Stats

**CCDTemperatureOK – UInt32 – R**

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

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

**StatDriverType – Enum – R**

<i>Standard</i>	The default network card driver is being used only
<i>Filter</i>	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.

For technical support, please contact [support@alliedvisiontec.com](mailto:support@alliedvisiontec.com).

For comments or suggestions regarding this document, please contact [info@alliedvisiontec.com](mailto:info@alliedvisiontec.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.

Copyright © 2013.

This document was prepared by the staff of Allied Vision Technologies Canada ("AVT") and is the property of AVT, which also owns the copyright therein. All rights conferred by the law of copyright and by virtue of international copyright conventions are reserved to AVT. This document must not be copied, or reproduced in any material form, either wholly or in part, and its contents and any method or technique available there from must not be disclosed to any other person whatsoever without the prior written consent of AVT.