Allied Vision

GigE Vision Cameras

GigE Features Reference

V4.0.0

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

2015-August-25

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

President/CEO: Frank Grube
Tax ID: DE 184383113

Headquarters:
Taschenweg 2a
D-07646 Stadtroda, Germany

Contents

Contact us

Connect with Allied Vision by function

http://www.alliedvision.com/en/meta-header/contact.html

Find an Allied Vision office or Allied Vision distributor

http://www.alliedvision.com/en/about-us/where-we-are.html

Email

info@alliedvision.com

support@alliedvision.com

Telephone

EMEA: +49 36428-677-0
The Americas: +1 978-225-2030
Asia-Pacific: +65 6634-9027
China: +86 (21) 64861133

Headquarters

Allied Vision Technologies GmbH
Taschenweg 2a
07646 Stadtroda
Germany

Tel: +49 (0)36428 6770
Fax: +49 (0)36428 677-28
President/CEO: Frank Grube
Registration Office: AG Jena HRB 208962

Introduction

This chapter includes:

- About this document
- Document history
- Layout styles and symbols used in this manual

About this document

The document describes the standard and advanced camera controls for Allied Vision GigE cameras as seen from the Vimba Viewer.

This document applies to all GigE Vision camera families and is divided into two main chapters:

- Chapter *GigE camera and driver features* describes the features for the following camera families using Allied Vision feature naming convention:

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

- Chapter *GigE IR & scientific camera and driver features* describes the features for Goldeye G and Goldeye G Cool cameras using the GenICam standard feature naming convention.

Further information available online

For more information on Allied Vision Cameras:

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

This is the master document for all Allied Vision GigE camera models. Some features are not available for all camera models.

Example:

- White balance is not available for monochrome cameras.

Some features are implemented in the cameras, but are not always available.

Examples:

- Color correction features are implemented in Manta, Mako G, and GT color cameras, but not the GB, GE, GC, GX cameras.
- Color Correction is supported in Manta, Mako G, and GT; but it is not available in color cameras if they are operated with Bayer pixel formats, but works if debayering is done within the camera.

Document history

Version	Date	Remarks
V1.0.0	2013-Jul-04	NEW MANUAL - RELEASE status
V1.0.1	2013-Sep-06	• Added the EF lens controls on page 20 • Added `ReverseX` control on page 70 • Updated `DefectMaskPixelEnable` feature • Updated controls in the `Statistics` feature • Updated controls in the `DeviceStatus` feature
V2.0.0	2014-Jul-22	• Added Chapter AVT GigE IR & scientific camera and driver features • Created Chapter AVT GigE camera and driver features by merging camera controls and driver controls chapters of V1.0.1 of this document • Added `BufferHandlingControl` and `StreamInformation` categories, applicable for Vimba V1.3 or higher • Replaced `GVCPHBInterval` with `GevHeartbeatTimeout` and `GevHeartbeatInterval`, applicable for Vimba V1.3 or higher • Update the following in Chapter AVT GigE camera and driver features - Updated `PixelFormat`, `Hue`, `Saturation`, and `ColorTransformationControl` - For Vimba Viewer v1.1.1 or higher, `GevDeviceMACAddress` is moved under `Info` - Updated `ChunkModeActive`, and `AcquisitionFrameRateAbs` - Added note on binning in `BinningHorizontal` and `BinningVertical` - Removed the EF lens controls from the document until the camera samples are available - Removed *`FrameTrigger`* from `SyncOutSource` on page 64
V2.0.1	2014-Aug-15	• Removed the unavailable pixel formats from the list for AVT GigE IR & scientific cameras • Removed EF lens information from the `ChunkModeActive` control
V2.0.2	2014-Oct-08	• Added background color to AVT GigE IR & scientific camera and driver features chapter to distinguish it from AVT GigE camera and driver features chapter. • Following changes are made in the AVT GigE camera and driver features chapter: - Updated `ChunkModeActive`, `BinningHorizontal`, `BinningVertical`, `DecimationHorizontal`, `DecimationVertical`, `PTP`, and `LUTControl`

Table 1: *Document history*

Version	Date	Remarks
V2.0.2 [Continued]	2014-Oct-08 [Continued]	- Moved `ReverseX` under `ImageMode` category - Added `ReverseY` - Removed `GainRaw` - Updated ExposureTimeAbs, `GainAuto`, and `Gain` - Added `ExposureTimeIncrement` - Removed the *`other`* option from `ExposureAuto` - Added *`TriggerWidth`* option for `ExposureMode`
V3.0.0	2015-Jan-15	• Updated Allied Vision logo • Updated `Statistics` category in both *GigE camera and driver features* and *GigE IR & scientific camera and driver features* chapters • Renamed: - Chapter 'AVT GigE camera and driver features' to *GigE camera and driver features* - Chapter 'AVT GigE IR & scientific camera and driver features' to *GigE IR & scientific camera and driver features* • Following changes are made in the *GigE camera and driver features* chapter: - Added `SensorShutterMode`, `BinningVerticalMode`, `BinningHorizontalMode`, and `DefectMaskEnable` - Updated `BinningHorizontal`and `BinningVertical` - Added *`PieceWiseLinearHDR`* option in `ExposureMode` - Added `ExposureTimePWL1`, `ExposureTimePWL1`, `ThresholdPWL1`, and `ThresholdPWL1` - Updated `ExposureTimeAbs`, `ExposureAuto`, `AcquisitionFrameRateAbs`, `GainAuto`, `IrisMode`, and `BalanceWhiteAuto` • Following changes are made in the *GigE IR & scientific camera and driver features* chapter: - Moved `BandwidthControlMode` under `DeviceControl` category - Added `DeviceFamilyName`, `DeviceFanMode`, `DeviceFanRpm`, `DeviceLinkHeartbeatTimeout`, and `DeviceFanSelector` under `DeviceControl` category - Updated `ExposureAuto`, `NUCDatasetDescription`, `SensorTemperatureControlState`, `SensorTemperatureSetpointValue`, and `GevHeartbeatTimeout` - Removed Line3 and Line 4 references from `EventData`, `EventSelector`, `TriggerSource`, `LineInSelector`, `LineOutSelector`, `LineOutSource`, `StrobeSource`, and `EventID` as these are not implemented in camera firmware - Added `GigEVision` category

Table 1: *Document history (Continued)*

Version	Date	Remarks
V3.0.0 [Continued]	2015-Jan-15 [Continued]	- Added `SensorOffsetX` and `SensorOffsetY` under `ImageFormatControl` category - Moved `StreamInformation` before `TransportLayerControl` - Removed `GVCPHBInterval` as it is replaced by `GevHeartbeatInterval` in Vimba 1.3 - Removed `GevHeartbeatTimeout` because it is replaced by `DeviceLinkHeartbeatTimeout` in camera firmware V2.04.03
V3.1.0	2015-Mar-10	• Added `EFLensControl` • Updated `DefectMaskEnable`, `PtpMode`, and `PtpStatus` • Updated `ChunkModeActive` and `SensorShutterMode`
V3.2.0	2015-Mar-20	• Replaced old links with new Allied Vision website links • Changed this documents name from 'GigE Camera and Driver Features' to 'GigE Features Reference' • Following changes are made in the *GigE IR & scientific camera and driver features* chapter: - Added `BackgroundCorrection` category - Added `IntegrationMode` control - Updated `NUCDatasetGain`, `NUCDatasetActiveGain`, and `SensorTemperatureSetpointSelector` - Updated `SensorTemperatureSetpointActive`, `NonImagePayloadSize`, and `SensorGain`
V4.0.0	2015-Aug-25	• Updated the document according to Allied Vision's new layout and brand guidelines • Added `GevIPConfigurationApply` feature in *GigE camera and driver features* chapter • Following changes are made in the *GigE IR & scientific camera and driver features* chapter: - Added `BinningHorizontal`, `BinningHorizontalMode`, `BinningVertical`, and `BinningVerticalMode` under `ImageFormatControl` category - Added `LUTControl` category - Added `GevIPConfigurationApply` feature

Table 1: *Document history (Continued)*

Manual conventions

To give this manual an easily understood layout and to emphasize important information, the following typographical styles and symbols are used:

Styles

Style	Function	Example
Bold	Highlighting important information	**control**
Courier	Camera feature	`Input`
Courier Italics	Possible feature values	*`Mode`*
Blue italics	Links	*Link*
Text in square brackets	Camera feature type: - Enum - Float - Integer - String - Command - Boolean - Register	***[Enum]***

Abbreviations

Abbreviation	Meaning
R/W	Feature is read/write
R/(W)	Feature is readable, and may be writable depending upon the user privilege level
R/C	Feature is read only and constant
R	Feature is read only and may change

Symbols

This symbol highlights important information.

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

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

Additional information

Allied Vision software

All software packages provided by Allied Vision 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

All software packages (including documentation and release notes) provided by Allied Vision can be downloaded at:

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

Third-party software

In general, third-party software provides increased functionality such as image processing and video recording. Vimba SDK is based on the GenICam standard. GenICam-based third-party software automatically connect with Vimba's transport layers. Additionally, Vimba includes the Cognex Adapter for VisionPro.

GigE camera and driver features

This chapter lists standard and advanced camera and driver controls, as seen from the Vimba Viewer, for the following camera families:

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

Acquisition

This group of controls relates to image acquisition.

AcquisitionAbort [Command]

Origin of feature: Camera

Software command to stop camera from receiving frame triggers and abort the current acquisition. A partially transferred image will be completed.

AcquisitionFrameCount [Integer] R/W

Origin of feature: Camera		
Range: [1–65535]	Default: ***1***	Unit: Frames

Defines 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`.

AcquisitionFrameRateAbs [Float] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: Frames per second

When `TriggerSelector` = ***`FrameStart`*** and either `TriggerMode` = ***`Off`*** or `TriggerSource` = ***`FixedRate`***, this control specifies the frame rate. Depending on the exposure duration, the camera may not achieve the frame rate set here.

- If `ExposureMode` = ***`Timed`***:
  Ensure [1/`ExposureTimeAbs`*] > `AcquisitionFrameRateAbs` to achieve target frame rate.
- If `ExposureMode` = ***`TriggerWidth`***:
  Ensure [1/(external trigger pulse width)] > `AcquisitionFrameRateAbs` to achieve target frame rate.
- If `ExposureMode` = ***`PieceWiseLinearHDR`***:
  Ensure the [1/`ExposureTimeAbs`*] > `AcquisitionFrameRateAbs` to achieve target frame rate.

* `ExposureTimeAbs` in seconds

AcquisitionFrameRateLimit [Float] R

Origin of feature: Camera	
Range: [Camera dependent]	Unit: Frames per second

The maximum frame rate possible for the current exposure duration and image format.

AcquisitionMode [Enum] R/W

Origin of feature: Camera

Determines the behavior of the camera when acquisition start is triggered.

Continuous	[Default] After an acquisition start event, the camera will continuously receive frame trigger events. See **TriggerSelector** and **TriggerSource** for more information.
SingleFrame	The camera will only deliver a single frame trigger event. Further trigger events will be ignored until acquisition is stopped and restarted
MultiFrame	The camera will acquire the number of images specified by **AcquisitionFrameCount**. Further trigger events will be ignored until acquisition is stopped and restarted
Recorder	The camera will continuously record images into the camera on-board memory, but will not send them to the host until an ***AcquisitionRecord*** trigger signal is received. Further ***AcquisitionRecord*** trigger events will be ignored until acquisition is stopped and restarted. Combined with the **RecorderPreEventCount** control, this feature is useful for returning any number of frames before a trigger event. When ***AcquisitionRecord*** trigger is received, the currently imaging/acquiring image will complete as normal, and then at least one more image will be taken. The memory is a circular buffer, that starts rewriting images once it is full. Its size is determined by **AcquisitionFrameCount**

AcquisitionStart [Command]

Origin of feature: Camera

Software command to start camera receiving frame triggers. Valid when **TriggerMode** = ***Off***. See **TriggerSelector** = ***FrameStart*** trigger.

AcquisitionStop [Command]

Origin of feature: Camera

Software command to stop camera from receiving frame triggers. Valid when **TriggerMode** = ***Off***. See **TriggerSelector** = ***FrameStart*** trigger.

RecorderPreEventCount [Integer] R/W

Origin of feature: Camera		
Range:[1–65535]	Default: ***0***	Unit: Frames

Valid when `AcquisitionMode` = ***`Recorder`***. The number of frames returned before the ***`AcquisitionRecord`*** trigger event, with `AcquisitionFrameCount` `minus` `RecorderPreEventCount` frames being returned after the ***`AcquisitionRecord`*** trigger event.

At least one image must be captured after the ***`AcquisitionRecord`*** trigger event, i.e., you cannot set `RecorderPreEventCount` = 1, and `AcquisitionFrameCount` = 1.

SensorShutterMode [Enum] R/W

Origin of feature: Camera

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 `ExposureTimeAbs`
`Rolling`	Each row is reset, exposed, and read out in succession from top to bottom of image. All pixels have same `ExposureTimeAbs`. 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, `ExposureTimeAbs` is the time from global reset to start of readout of top row. Subsequent rows will have a longer exposure time (`ExposureTimeAbs` + row readout time × row number). This mode offers enhanced SNR/dynamic range with no motion blur, which is useful for strobe applications

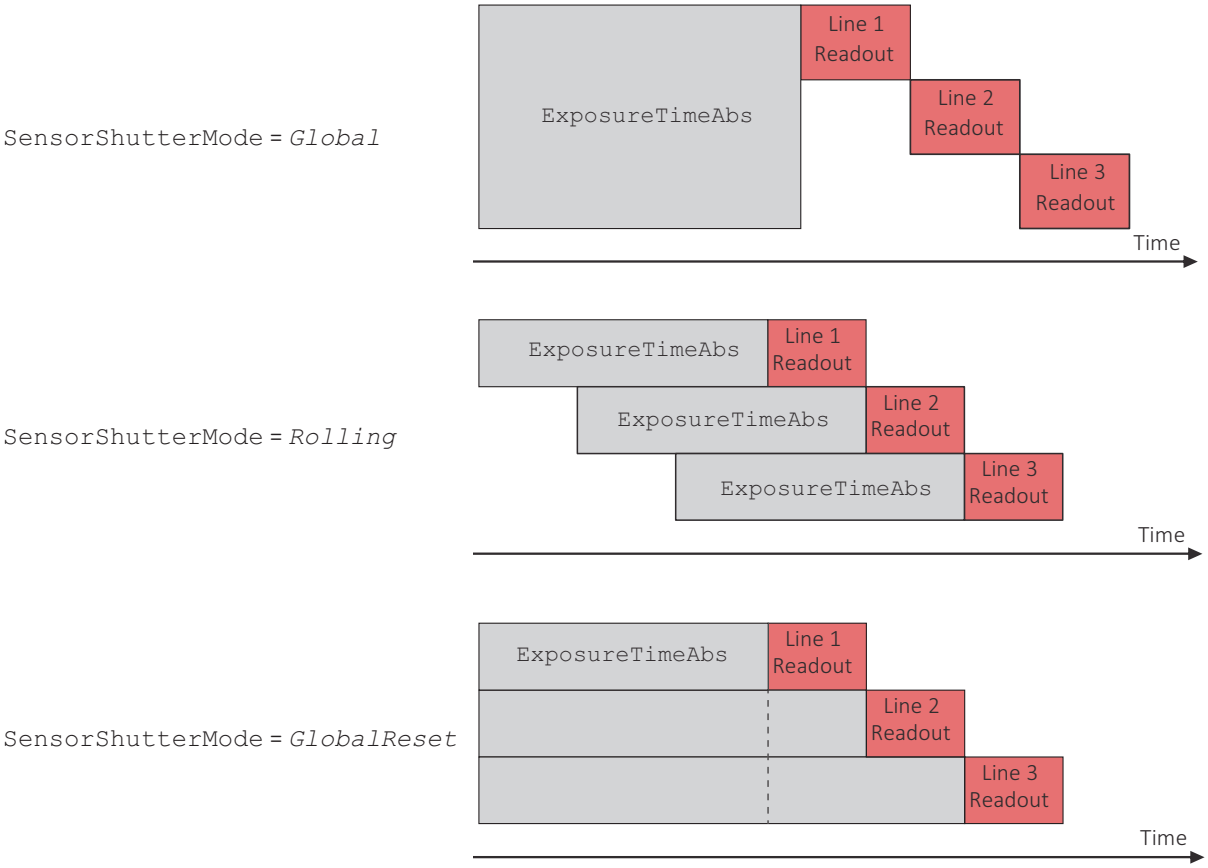

***Figure 1**: Illustration showing different sensor shutter modes*

Trigger

This group of controls relates to how an image frame is initiated or triggered.

TriggerActivation [Enum] R/W

Origin of feature: Camera

Type of activation, for hardware triggers. This controls edge/level and polarity sensitivities.

RisingEdge	[Default] Rising edge trigger
FallingEdge	Falling edge trigger
AnyEdge	Rising or falling edge
LevelHigh	Active high signal
LevelLow	Active low signal

TriggerDelayAbs [Float] R/W

Origin of feature: Camera		
Range:[0 – Camera dependent]	Default: ***0***	Unit: µs

Start-of-image can be delayed to begin some time after a trigger event is received by the camera. This feature is valid only when `TriggerSource` is set to external trigger (i.e. ***`Line1`***, ***`Line2`***). This control is a common trigger to sync with a strobe lighting source, which will inherently have some fixed setup time.

TriggerMode [Enum] R/W

Origin of feature: Camera

Enables or disables trigger set in `TriggerSelector`.

`Off`	Trigger disabled
`On`	[Default] Trigger enabled

iIf `TriggerMode` = ***`Off`*** and `TriggerSelector` = ***`FrameStart`***, images triggered in ***`FixedRate`*** at `AcquisitionFrameRateAbs`.

TriggerOverlap [Enum] R/W

Origin of feature: Camera

Permitted window of trigger activation, relative to previous frame. Does not work with software triggering. Only external.

`Off`	[Default] Any external trigger received before a high ***`FrameTriggerReady`*** signal is ignored
`PreviousFrame`	Any external trigger received before ***`FrameTriggerReady`*** is latched and used to trigger the next frame

TriggerSelector [Enum] R/W

Origin of feature: Camera

Selects a trigger, then use the controls {`TriggerMode`, `TriggerSoftware`, `TriggerSource`, `TriggerActivation`, `TriggerOverlap`, `TriggerDelayAbs`} to setup and read the trigger features.

`FrameStart`	[Default] The trigger which starts each image (when acquisition is running)
`AcquisitionStart`	The trigger which starts the acquisition process
`AcquisitionEnd`	The trigger which ends the acquisition process
`AcquisitionRecord`	The trigger which initiates the sending of `AcquisitionFrameCount` number of recorded images from the camera on-board memory to the host

TriggerSoftware [Command]

Origin of feature: Camera

Triggers an image. Valid when `TriggerSource` = ***`Software`***.

TriggerSource [Enum] R/W

Origin of feature: Camera

Determines how an image frame is initiated within an acquisition stream.

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

`Freerun`	[Default] Camera runs at maximum supported frame rate depending on the exposure time and region of interest (ROI) size
`Line1`	External trigger ***`Line1`***
`Line2`	External trigger ***`Line2`***
`Line3`	External trigger ***`Line3`***
`Line4`	External trigger ***`Line4`***
`FixedRate`	Camera self-triggers at a fixed frame rate defined by `AcquisitionFrameRateAbs`
`Software`	Software initiated image capture

BufferHandlingControl

StreamAnnounceBufferMinimum [Integer] R/C

Origin of feature: Driver

Display name: Stream Announce Buffer Minimum

Vimba V1.3 or later Minimal number of buffers to announce to enable selected acquisition mode.

StreamAnnouncedBufferCount [Integer] R

Origin of feature: Driver

Display name: Stream Announced Buffer Count

Vimba V1.3 or later Number of announced (known) buffers on this stream.

StreamBufferHandlingMode [Enum] R/W

Origin of feature: Driver

Display name: Stream Buffer Handling Mode

Vimba V1.3 or later Available buffer handling modes of this stream.

Controls

BlackLevelControl

BlackLevel [Float] R/W

Origin of feature: Camera	
Range: [0–255.75]	Default: *0*

Black level value. Setting the `Gain` does not change the `BlackLevel`.

BlackLevelSelector [Enum] R/W

Origin of feature: Camera	
All	[Default] `BlackLevel` will be applied to all channels or taps

CCDTemperatureOK [Integer] R

Origin of feature: Camera

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

0	[Default] The CCD sensor may be 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

ColorTransformationControl

This section describes features related to color transformations in the Allied Vision GigE color cameras. The following controls are only valid when using on-camera interpolated *`PixelFormats`*.

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 to change the coefficients of the 3x3 matrix.

$$\begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} Gain00 & Gain01 & Gain02 \\ Gain10 & Gain11 & Gain12 \\ Gain20 & Gain21 & Gain22 \end{bmatrix} \times \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix}$$

ColorTransformationMode [Enum] R/W

Origin of feature: Camera

Selects the mode for the color transformation.

Off	[Default] No color transformation
Manual	Manually set **ColorTransformationValue** matrix coefficients
Temp6500K	Colors optimized for a surrounding color temperature 6500 K

ColorTransformationSelector [Enum] R/W

Origin of feature: Camera
Possible values: ***RGBtoRGB***

Selects which color transformation module is controlled by the various color transformation features.

ColorTransformationValue [Float] R/W

Origin of feature: Camera	
Range: [-2 to 2]	Default: ***1***

Represents the value of the selected gain factor or offset inside the transformation matrix.

ColorTransformationValueSelector [Enum] R/W

Origin of feature: Camera

Selects the gain factor or offset of the transformation matrix if **ColorTransformationMode** = ***Manual***.

Gain00	[Default] Red contribution to the red pixel (multiplicative factor)
Gain01	Green contribution to the red pixel (multiplicative factor)
Gain02	Blue contribution to the red pixel (multiplicative factor)
Gain10	Red contribution to the green pixel (multiplicative factor)
Gain11	Green contribution to the green pixel (multiplicative factor)
Gain12	Blue contribution to the green pixel (multiplicative factor)
Gain20	Red contribution to the blue pixel (multiplicative factor)
Gain21	Green contribution to the blue pixel (multiplicative factor)
Gain22	Blue contribution to the blue pixel (multiplicative factor)

DSPSubregion

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 [Integer] R/W

Origin of feature: Camera	
Range: [0 – Sensor height]	Default: ***Sensor height***

Defines the bottom edge of the DSP subregion.

DSPSubregionLeft [Integer] R/W

Origin of feature: Camera	
Range: [0 – Sensor width]	Default: ***0***

Defines the left edge of the DSP subregion.

DSPSubregionRight [Integer] R/W

Origin of feature: Camera	
Range: [0 – Sensor width]	Default: ***Sensor width***

Defines the right edge of the DSP subregion.

DSPSubregionTop [Integer] R/W

Origin of feature: Camera	
Range: [0 – Sensor height]	Default: ***0***

Defines the top edge of the DSP subregion.

EdgeFilter [Enum] R/W

Origin of feature: Camera

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

Smooth2	Most blur
Smooth1	Slight blur
Off	[Default] No blur or sharpness applied
Sharpen1	Slight sharp
Sharpen2	Most sharp

`EdgeFilter` feature is applicable only to color models/Manta cameras except dual-tap camera models.

DefectMaskEnable [Boolean] R/W

Origin of feature: Camera

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

If `BinningHorizontal`, `BinningVertical`, `DecimationHorizontal`, or `DecimationVertical` is set greater than 1, `DefectMaskEnable` is set to `False`.

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.

See the modular concept document, or contact your Allied Vision 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

Origin of feature: Camera

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

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

EFLensControl

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

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

EFLensFStop

EFLensFStopCurrent [Float] R/W

Origin of feature: Camera	
Range: [`EFLensFStopMin` - `EFLensFStopMax`]	Unit: µs

Current F-stop number or aperture of the EF lens.

EFLensFStopDecrease [Command]

Origin of feature: Camera

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

EFLensFStopIncrease [Command]

Origin of feature: Camera

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

EFLensFStopMax [Float] R

Origin of feature: Camera	
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

Origin of feature: Camera	
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

Origin of feature: Camera	
Range: [1–8]	Unit: F-Stop/8

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

EFLensFocus

EFLensFocusCurrent [Integer] R/W

Origin of feature: Camera

Range: [`EFLensFocusMin` – `EFLensFocusMax`]

Current focus setting.

EFLensFocusDecrease [Command]

Origin of feature: Camera

Decrease/shorten focus distance by `EFLensFocusStepSize`.

EFLensFocusIncrease [Command]

Origin of feature: Camera

Increase/lengthen focus distance by `EFLensFocusStepSize`.

EFLensFocusMax [Integer] R

Origin of feature: Camera

Default: ***`Lens dependent`***

Maximum/farthest possible focus setting.

EFLensFocusMin [Integer] R

Origin of feature: Camera

Default: ***`Lens dependent`***

Minimum/nearest possible focus setting.

EFLensFocusStepSize [Integer] R/W

Origin of feature: Camera

Range: [1 – Lens dependent] Default: ***`10`***

Size of increments/decrements in `EFLensFocusCurrent` when using `EFLensFocusIncrease` and `EFLensFocusDecrease` commands, respectively.

EFLensFocusSwitch [Enum] R

Origin of feature: Camera

Current position of lens AF/MF switch.

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

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

EFLensInitialize [Command]

Origin of feature: Camera

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

EFLensStatus

EFLensID [Integer] R

Origin of feature: Camera

Identification value of the attached EF lens.

EFLensLastError [Enum] R

Origin of feature: Camera

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

Origin of feature: Camera

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

EFLensZoom

EFLensZoomCurrent [Integer] R

Origin of feature: Camera	
Range: [**EFLensZoomMin** – **EFLensZoomMax**]	Units: mm

Current focal length of the EF lens.

EFLensZoomMax [Integer] R

Origin of feature: Camera	
Default: ***Lens dependent***	Units: mm

Maximum focal length of the EF lens.

EFLensZoomMin [Integer] R

Origin of feature: Camera	
Default: ***Lens dependent***	Units: mm

Minimum focal length of the EF lens.

Exposure

ExposureAuto [Enum] R/W

Origin of feature: Camera

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.

Off	[Default] The automatic mode is Off
Once	Valid when `ExposureMode` = ***Timed*** or `PieceWiseLinearHDR`. Auto-exposure occurs until target is achieved, then `ExposureAuto` returns to ***Off***
Continuous	Valid when `ExposureMode` = ***Timed*** or `PieceWiseLinearHDR`. The exposure time will vary continuously according to the scene illumination. The auto exposure function operates according to the `ExposureAuto` and `DSPSubregion` controls

If using `ExposureAuto` = ***Continuous***, and `GainAuto` = ***Continuous*** 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.

You can configure the auto exposure feature to respond only to a subregion within the image scene. This subregion can be configured with the `DSPSubregion` feature.

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

ExposureAutoControl

ExposureAutoAdjustTol [Integer] R/W

Origin of feature: Camera		
Range: [0–50]	Default: ***5***	Unit: Percent

Tolerance in variation from `ExposureAutoTarget` in which the auto exposure algorithm will not respond. It can be used to limit exposure setting changes to only larger variations in scene lighting.

ExposureAutoAlg [Enum] R/W

Origin of feature: Camera

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

ExposureAutoMax [Integer] R/W

Origin of feature: Camera		
Range: [Camera dependent]	Default: ***500000***	Unit: µs

The upper bound to the exposure setting in auto exposure mode. This is useful in situations where frame rate is important. This value would normally be set to something less than (as a rough estimate)$1x10^6$/(desired frame rate).

ExposureAutoMin [Integer] R/W

Origin of feature: Camera		
Range: [Camera dependent]	Default: ***Camera dependent***	Unit: µs

The lower bound to the exposure setting in auto exposure mode.

ExposureAutoOutliers [Integer] R/W

Origin of feature: Camera		
Range: [0–1000]	Default: ***0***	Unit: 0.01% i.e. 1000 = 10%

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

ExposureAutoRate [Integer] R/W

Origin of feature: Camera		
Range: [1–100]	Default: ***100***	Unit: Percent

The rate at which the auto exposure function changes the exposure setting. 100% is auto exposure adjustments running at full speed, and 50% is half speed.

ExposureAutoTarget [Integer] R/W

Origin of feature: Camera		
Range: [1–100]	Default: ***50***	Unit: 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

Origin of feature: Camera	
Timed	[Default] Camera exposure time is set by `ExposureTimeAbs`
TriggerWidth	Camera exposure time is controlled by external trigger pulse on ***Line1*** or ***Line2***. In order for this feature to work, `TriggerSelector` = ***FrameStart*** and `TriggerSource` must be set to ***Line1*** or ***Line2***
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 `ExposureTimeAbs`. HDR sub-exposures are set using `ExposureTimePWL1` and `ExposureTimePWL2`

Control for exposure duration.

ExposureTimeAbs [Float] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: µs

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

`ExposureTimeAbs` depends on `ExposureMode` as follows:

- `ExposureMode` = ***Timed***: `ExposureTimeAbs` is sensor integration time.
- `ExposureMode` = ***TriggerWidth***: `ExposureTimeAbs` is ignored.
- `ExposureMode` = ***PieceWiseLinearHDR***: `ExposureTimeAbs` is the full sensor integration time. See `ExposureTimePWL1` and `ExposureTimePWL2` for setting `ThresholdPWL` exposure durations.

ExposureTimeIncrement [Float] R/C

Origin of feature: Camera	
Range: [Camera dependent]	Unit: µs

Increment/resolution of the exposure time in microseconds.

ExposureTimePWL1 [Float] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: µs

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

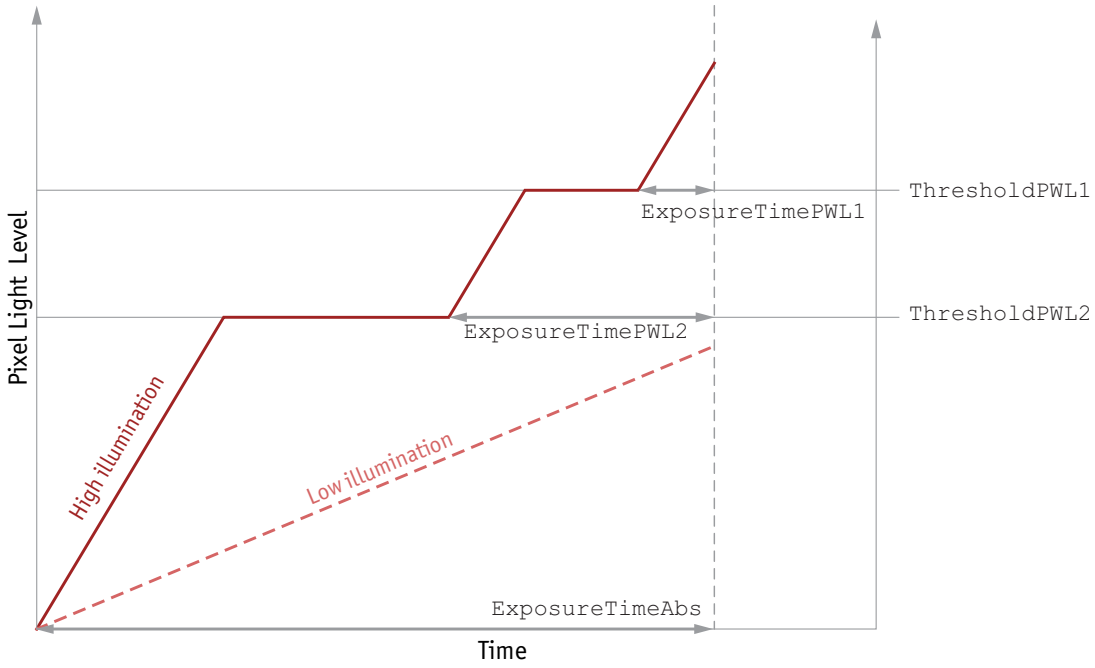

Figure 2**: HDR sub exposures and thresholds when* `ExposureMode` = ***`PieceWiseLinearHDR`

ExposureTimePWL2 [Float] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: µs

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

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

ThresholdPWL1 [Integer] R/W

Origin of feature: Camera	
Range: [0–63]	Default: ***63***

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

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

ThresholdPWL2 [Integer] R/W

Origin of feature: Camera	
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.

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

Shutter [Enum] R/W

Origin of feature: Camera

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 ***`LineIn1`***
`SyncIn2`	Enables or disables the mechanical shutter dependent on the level of ***`LineIn2`***
`SyncIn3`	Enables or disables the mechanical shutter dependent on the level of ***`LineIn3`***
`SyncIn4`	Enables or disables the mechanical shutter dependent on the level of ***`LineIn4`***
`SyncIn5`	Enables or disables the mechanical shutter dependent on the level of ***`LineIn5`***

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.

GainControl/Gain

This feature controls the gain settings applied to the sensor.

Gain [Float] R/W

Origin of feature: Camera		
Range: [Camera dependent]	Default: ***0***	Unit: 1 dB

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

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.

GainAuto [Enum] R/W

Origin of feature: Camera

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 to 3 frames for the algorithm to stabilize.

- Auto algorithm adjusts using 1 dB gain steps.
- The camera must be acquiring images in order for the auto algorithm to update.

Off	[Default] The automatic mode is Off
Once	Valid when `ExposureMode` = ***Timed*** or `PieceWiseLinearHDR`. Auto-gain occurs until target is achieved, then `GainAuto` returns to ***Off***
Continuous	Valid when `ExposureMode` = ***Timed*** or `PieceWiseLinearHDR`. The gain will vary continuously according to the scene illumination. The auto exposure function operates according to the `ExposureAutoControl` and `DSPSubregion` controls

If using `ExposureAuto` = ***Continuous***, and `GainAuto` = ***Continuous*** 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.

You can configure the auto gain feature to respond only to a subregion within the image scene. This subregion can be configured with the `DSPSubregion` feature.

GainAutoControl

GainAutoAdjustTol [Integer] R/W

Origin of feature: Camera		
Range: [0–50]	Default: ***5***	Unit: Percent

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

GainAutoMax [Float] R/W

Origin of feature: Camera	
Range: [0 – Camera dependent]	Unit: dB

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

GainAutoMin [Float] R/W

Origin of feature: Camera		
Range: [0 – Camera dependent]	Default: ***0***	Unit: dB

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

GainAutoOutliers [Integer] R/W

Origin of feature: Camera		
Range: [1–1000]	Default: ***0***	Unit: 0.01%, i.e. 1000 = 10%

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

GainAutoRate [Integer] R/W

Origin of feature: Camera		
Range: [1–100]	Default: ***100***	Unit: Percent

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

GainAutoTarget [Integer] R/W

Origin of feature: Camera		
Range: [0–100]	Default: ***50***	Unit: Percent

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

GainSelector [Enum] R/W

Origin of feature: Camera	
`All`	[Default] Gain will be applied to all channels or taps

Gamma [Float] R/W

Origin of feature: Camera		
Range: Camera dependent	Default: `1.00`	Unit: Output = (Input)^{Gamma}

Nonlinear brightness control. Applies gamma value to the raw sensor signal (via LUT).

`1.00`	`Gamma` OFF (no Gamma correction)
Values other than `1.00`	`Gamma` ON

For Manta type A

If `Gamma` is ON, LUT 1 is used to do the gamma transform. The original LUT values will be stored temporarily. If `Gamma` is ON, and you read out LUT1: you only get stored LUT values but not `Gamma` values. In general, Gamma values cannot be read out.
If `Gamma` is OFF, LUT position 1 contains optional user defined LUT values.

Manta type B, Mako G, and Prosilica GT cameras have a stand-alone gamma function which does not share resources with LUTs.

Hue [Float] R/W

Origin of feature: Camera		
Range: Camera dependent	Default: `0.00`	Unit: Degrees

Alters color of image without altering white balance. Takes float input, although rounds to integer. Only valid when using on-camera interpolated `PixelFormats`.

IODMode [Enum] R/W

Origin of feature: Camera

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, Bigeye G-283B Cool, and Bigeye G-1100B Cool achieve maximum frame rate in continuous mode only.
IOD	[Default] Enables IOD mode. In this mode the camera needs an external trigger signal or a timer driven internal exposure signal
LineIn1	The camera is switched between ***IOD*** and ***Continuous*** mode, dependent on the level of ***LineIn1***
LineIn2	The camera is switched between ***IOD*** and ***Continuous*** mode, dependent on the level of ***LineIn2***
LineIn3	The camera is switched between ***IOD*** and ***Continuous*** mode, dependent on the level of ***LineIn3***
LineIn4	The camera is switched between ***IOD*** and ***Continuous*** mode, dependent on the level of ***LineIn4***
LineIn5	The camera is switched between ***IOD*** and ***Continuous*** mode, dependent on the level of ***LineIn5***

If ***Continuous*** mode is activated, no external exposure signal is allowed. Set `TriggerSelector` to ***FrameStart*** and `TriggerSource` to an unused external trigger Line.

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 `GainAuto` = ***Continuous***, or `ExposureAuto` = ***Continuous***, to `IrisMode` = ***Video/DCIris/PIrisAuto*** results in undefined, “race to target” behavior.

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

IrisAutoTarget [Integer] R/W

Origin of feature: Camera		
Range: [0–100]	Default: ***50***	Unit: 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

Origin of feature: Camera

Sets the auto iris mode. Valid when `ExposureMode` = ***`Timed`*** or `PieceWiseLinearHDR`.

`Disabled`	[Default] Disable auto iris
`Video`	Enable video iris. Video-type lenses only
`VideoOpen`	Fully open a video iris. Video-type lenses only
`VideoClose`	Full close a video iris. Video-type lenses only
`PIrisAuto`	Enable precise auto iris. P-Iris lenses only
`PIrisManual`	Manually control iris via `LensPIrisPosition` feature. P-Iris lenses only
`DCIris`	Enable DC auto iris. DC-Iris lenses only

IrisVideoLevel [Integer] R

Origin of feature: Camera		
Range: [0–150]	Default: ***0***	Unit: mV pp

Current video iris level, which is the strength of the video signal coming from the camera. Dependent on lens type.

Lens type	Range	Description
Video-type lenses	[0–150]	Reference voltage. This value should fall between `IrisVideoLevelMin` and `IrisVideoLevelMax`
P-Iris lenses	[0–100]	Attempts to match `IrisAutoTarget`
DC-Iris lenses	[0–100]	Attempts to match `IrisAutoTarget`

IrisVideoLevelMax [Integer] R/W

Origin of feature: Camera		
Range: [0–150]	Default: ***Camera dependent***	Unit: 10 mV [Manta: 13.2 mV]

Video-type lenses only. Limits the maximum driving voltage for closing the lens iris. Typically, this will be 150; however, it may vary depending on the lens reference voltage. A lower minimum value slows the adjustment time but prevents excessive overshoot.

IrisVideoLevelMin [Integer] R/W

Origin of feature: Camera		
Range: [0–150]	Default: ***Camera dependent***	Unit: 10 mV [Manta: 13.2 mV]

Video-type lenses only. Limits the minimum driving voltage for opening the lens iris. A higher minimum value slows the adjustment time but prevents excessive overshoot.

LensDCIris

DC-Iris lenses only.

LensDCDriveStrength [Integer] R/W

Origin of feature: Camera	
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 optimal 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.

For a list of P-Iris supported lenses:

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

LensPIrisFrequency [Integer] R/W

Origin of feature: Camera		
Range: [0–1000]	Default: ***100***	Unit: Hz

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

LensPIrisNumSteps [Integer] R/W

Origin of feature: Camera	
Range: [0–1023]	Default: ***50***

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

LensPIrisPosition [Integer] R/W

Origin of feature: Camera	
Range: [0–1022]	Default: ***50***

Iris/aperture position. Manually control iris in ***PIrisManual*** mode, or read back iris position in ***PIrisAuto*** mode. ***0*** represents fully open and ***1022*** represents fully closed position. Values greater than **LensPIrisNumSteps** are ignored/not written.

LensDrive

Open loop DC 3 axis lens control.

LensDriveCommand [Enum] R/W

Origin of feature: Camera

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

Stop	No action
IrisTimedOpen	Open lens iris
IrisTimedClos	Close lens iris
FocusTimedNea	Shorten working distance
FocusTimedFar	Lengthen working distance
ZoomTimedIn	Zoom in
ZoomTimedOut	Zoom out

LensDriveDuration [Integer] R/W

Origin of feature: Camera		
Range: [0–5000]	Default: ***0***	Unit: ms

Duration of timed lens commands.

LensVoltage [Integer] R

Origin of feature: Camera		
Range: [0–12000]	Default: ***0***	Unit: mV

Reports the lens power supply voltage.

LensVoltageControl [Integer] R/W

Origin of feature: Camera		
Range: [0–1200012000]	Default: `0`	Unit: mV * 100001

Lens power supply voltage control. See lens documentation for appropriate voltage level. Set desired lens voltage in mV*100001. This is done to prevent users inadvertently setting an inappropriate voltage, possibly damaging the lens. If a bad value is written this control resets to `0`.

LUTControl

Use of a 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.

Color cameras only:

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

LUTEnable [Boolean] R/W

Origin of feature: Camera	
Possible values: `True`, `False`	Default: `False`

Activates or deactivates the selected LUT.

LUTIndex [Integer] R/W

Origin of feature: Camera	
Range: [0 – ($2^{LUTBitDepthIn}$ - 1)]	Default: `0`

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

LUTInfo

This control provides active LUT information.

LUTAddress [Integer] R/C

Origin of feature: Camera

Indicates location of memory, when LUT is loaded.

LUTBitDepthIn [Integer] R/C

Origin of feature: Camera
Display name: LUTBitLengthIn

Bit depth of the input value of the LUT block.

LUTBitDepthOut [Integer] R/C

Origin of feature: Camera
Display name: LUTBitLengthOut

Bit depth of the output value of the LUT block.

LUTSizeBytes [Integer] R/C

Origin of feature: Camera
Display name: LUTSize

Memory size of the active LUT.

LUTLoadAll / LUTLoad [Command]

Origin of feature: Camera

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

LUTMode [Enum] R/W

Origin of feature: Camera

Selects on which pixels the selected LUT (depending on `LUTSelector`) 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

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

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

LUTSaveAll / LUTSave [Command]

Origin of feature: Camera

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

With `UserSets` control (`UserSetSave` command) you cannot save the contents of the LUT.

LUTSelector [Enum] R/W

Origin of feature: Camera	
Possible values: ***LUT1***, ***LUT2***, ***LUT3***, ***LUT4***, ***LUT5***	Default: ***LUT1***

Selects which look-up table is used. These LUTs are camera specific.

LUTValue [Integer] R/W

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

Returns or sets the value at entry `LUTIndex`.

NirMode [Enum] R/W

Origin of feature: Camera

Select 3 different NIR modes. The modes differ in quantum efficiency, frame rates, and anti-blooming characteristics

Off	[Default] `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	Cannot 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, `ExposureTimeAbs`/4) **Anti-blooming characteristics**: • Very good if, `ExposureAuto` = ***Off*** • Adaptively reduced if, `ExposureTimeAbs` < 13200 µs or `ExposureAuto` = Other **Usage**: Best suited for medium length exposure times and 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, when high frame rate is desired

Saturation [Float] R/W

Origin of feature: Camera
Range: 0.00–2.00]

Alters color intensity. Only valid when using on-camera interpolated ***PixelFormats***.

0.00	Monochrome
1.00	[Default] Default saturation
2.00	Maximum possible saturation that can be applied

SubstrateVoltage

VsubValue [Integer] R

Origin of feature: Camera	
Range: [Camera dependent]	Unit: mV

CCD substrate voltage. Optimized at factory for each sensor.

Whitebalance

BalanceRatioAbs [Float] R/W

Origin of feature: Camera

Adjusts the gain of the channel selected in the `BalanceRatioSelector`. `BalanceRatioAbs` = ***1.00*** means no gain is applied.

The green channel gain is always 1.00, as this is the luminance/reference channel. To increase/decrease green, decrease/increase red and blue accordingly.

BalanceRatioSelector [Enum] R/W

Origin of feature: Camera	
Possible values: ***Red***, ***Blue***	Default: ***Red***

Select the red or blue channel to adjust with `BalanceRatioAbs`.

BalanceWhiteAuto [Enum] R/W

Origin of feature: Camera

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.

You can configure the auto white balance feature to respond only to a subregion within the image scene. This subregion can be configured with the `DSPSubregion` feature.

Off	[Default] Auto white balance is off. White balance can be adjusted directly by changing the `BalanceRatioSelector` and `BalanceRatioAbs`
Once	Valid when `ExposureMode` = ***Timed*** or `PieceWiseLinearHDR`. A single iteration of the auto white balance algorithm is run, and then `BalanceWhiteAuto` returns to ***Off***. The ***Once*** function operates according to the `ExposureAuto` and `DSPSubregion` controls
Continuous	Valid when `ExposureMode` = ***Timed*** or `PieceWiseLinearHDR`. White balance will continuously adjust according to the current scene. The ***continuous*** function operates according to the `ExposureAuto` and `DSPSubregion` controls

BalanceWhiteAutoControl

BalanceWhiteAutoAdjustTol [Integer] R/W

Origin of feature: Camera		
Range: [0–50]	Default: ***5***	Unit: Percent

Tolerance allowed from the ideal white balance values, within which the auto white balance does not run. It is used to limit white balance setting changes to only larger variations in color.

BalanceWhiteAutoRate [Integer] R/W

Origin of feature: Camera		
Range: [0–100]	Default: ***100***	Unit: Percent

Rate of white balance adjustments, from 1 (slowest) to 100 (fastest). It is used to slow the rate of color balance change so that only longer period fluctuations affect color.

DeviceStatus

DeviceTemperature [Float] R

Origin of feature: Camera		
Unit: Degree Celsius	Resolution: 0.031	Accuracy: ±1 °C

Camera internal temperature.

DeviceTemperatureSelector [Enum] R/W

Origin of feature: Camera

Possible values: `Main`, `Sensor`

Selects the site whose temperature is reported by `DeviceTemperature`.

EventControl

The following table lists all the events supported by the camera:

EventData

Origin of all features: Camera

EventAcquisitionEndFrameID	*EventLine2FallingEdgeTimestamp*
EventAcquisitionEndTimestamp	*EventLine2RisingEdgeFrameID*
EventAcquisitionRecordTriggerFrameID	*EventLine2RisingEdgeTimestamp*
EventAcquisitionRecordTriggerTimestamp	*EventLine3FallingEdgeFrameID*
EventAcquisitionStartFrameID	*EventLine3FallingEdgeTimestamp*
EventAcquisitionStartTimestamp	*EventLine3RisingEdgeFrameID*
EventErrorFrameID	*EventLine3RisingEdgeTimestamp*
EventErrorTimestamp	*EventLine4FallingEdgeFrameID*
EventExposureEndFrameID	*EventLine4FallingEdgeTimestamp*
EventExposureEndTimestamp	*EventLine4RisingEdgeFrameID*
EventFrameTriggerFrameID	*EventLine4RisingEdgeTimestamp*
EventFrameTriggerTimestamp	*EventOverflowFrameID*
EventLine1FallingEdgeFrameID	*EventOverflowTimestamp*
EventLine1FallingEdgeTimestamp	*EventPtpSyncLockedFrameID*
EventLine1RisingEdgeFrameID	*EventPtpSyncLockedTimestamp*
EventLine1RisingEdgeTimestamp	*EventPtpSyncLostFrameID*
EventLine2FallingEdgeFrameID	*EventPtpSyncLostTimestamp*

EventID

Origin of all features: Camera

`EventAcquisitionStart [Integer] R/C`	*40000*
`EventAcquisitionEnd [Integer] R/C`	*40001*
`EventFrameTrigger [Integer] R/C`	*40002*
`EventExposureEnd [Integer] R/C`	*40003*
`EventAcquisitionRecordTrigger [Integer] R/C`	*40004*
`EventPtpSyncLost [Integer] R/C`	*40005*
`EventPtpSyncLocked [Integer] R/C`	*40006*
`EventLine1RisingEdge [Integer] R/C`	*40010*
`EventLine1FallingEdge[Integer] R/C`	*40011*
`EventLine2RisingEdge [Integer] R/C`	*40012*
`EventLine2FallingEdge [Integer] R/C`	*40013*
`EventLine3RisingEdge [Integer] R/C`	*40014*
`EventLine3FallingEdge[Integer] R/C`	*40015*
`EventLine4RisingEdge [Integer] R/C`	*40016*
`EventLine4FallingEdge [Integer] R/C`	*40017*
`EventFrameTriggerReady [Integer] R/C`	*40018*
`EventOverflow [Integer] R/C`	*65534*
`EventError [Integer] R/C`	*65535*

If you use the message channel for event notification, you are always subscribed to `EventOverflow` and `EventError` events.

There is no mechanism to detect the loss of events during transportation. If mis-configured, cameras may produce lots of events—more than a PC can handle.

EventNotification [Enum] R/W

Origin of feature: Camera	
Possible values: *`On`*, *`Off`*	Default: *`Off`*

Activates event notification on the GigE Vision message channel.

EventSelector [Enum] R/W

Origin of feature: Camera

Selects a specific event to be enabled or disabled using `EventNotification`. Possible values are listed as following:

AcquisitionStart [Default]	*AcquisitionEnd*
FrameTrigger	*ExposureEnd*
AcquisitionRecordTrigger	*PtpSyncLost*
PtpSyncLocked	*Line1RisingEdge*
Line1FallingEdge	*Line2RisingEdge*
Line2FallingEdge	*Line3RisingEdge*
Line3FallingEdge	*Line4RisingEdge*
Line4FallingEdge	*FrameTriggerReady*

EventsEnable1 [Integer] R/W

Origin of feature: Camera

Default: *0*

Bit field of all events. For example:

Bit 1	EventAcquisitionStart
Bit 2	EventAcquisitionEnd
Bit 3	EventFrameTrigger
Bit 19	EventFrameTriggerReady

This is an alternative to setting each event individually using the `EventNotification` and `EventSelector` method.

GigE

BandwidthControlMode [Enum] R/W

Origin of feature: Camera

Selects the desired mode of bandwidth control.

StreamBytesPerSecond	[Default] See the `StreamBytesPerSecond` feature for more information
SCPD	Stream channel packet delay expressed in timestamp 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

Origin of feature: Camera	
Possible values: `True`, `False`	Default: `False`

Enables camera to send GigE Vision Standard Protocol chunk data with an image. `ChunkModeActive` is read-only during acquisition. 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.

Configuration

GevIPConfigurationApply [Command]

Origin of feature: Driver
Display name: IP Configuration Apply

Apply the IP configuration mode selected by `GevIPConfigurationMode`.

GevIPConfigurationMode [Enum] R/W

Origin of feature: Driver
Display name: IP Configuration Mode
Possible values: *`LLA`*, *`DHCP`*, *`Persistent`*

Current IP configuration mode.

Current

GevCurrentDefaultGateway [Integer] R

Origin of feature: Driver
Display name: Current Default Gateway

IP address of the default Gateway of the device.

GevCurrentIPAddress [Integer] R

Origin of feature: Driver
Display name: Current IP Address

Current IP address of the device.

GevCurrentSubnetMask [Integer] R

Origin of feature: Driver
Display name: Current Subnet Mask

Current Subnet Mask of the device.

GVCP

Definition

GVCP = GigE Vision Control Protocol
Allied Vision GigE cameras have a sophisticated real time resend mechanism that ensures a high degree of data integrity.

GVCPCmdRetries [Integer] R/W

Origin of feature: Driver	
Display name: Command Retries	
Range:[1–9]	Default: ***5***

Controls the maximum number of resend requests that the host will attempt when trying to recover a lost packet.

GVCPCmdTimeout [Integer] R/W

Origin of feature: Driver		
Display name: Command Timeout		
Range:[100–1000]	Default: ***250***	Unit: ms

Timeout waiting for an answer from the device.

GevHeartbeatInterval [Integer] R/W

Origin of feature: Driver		
Display name: Heartbeat Interval		
Range:[200–1450]	Default: ***1450***	Unit: ms

Vimba V1.3 or later

The driver sends heartbeat packets to the camera every `GevHeartbeatInterval` milliseconds.

GevHeartbeatTimeout [Integer] R/W

Origin of feature: Driver		
Display name: Heartbeat Timeout		
Range:[500–10000]	Default: ***3000***	Unit: ms

Vimba V1.3 or later

The driver sends heartbeat packets to the camera. If a heartbeat packet is not received within `GevHeartbeatTimeout`, 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.

GVCPHBInterval [Integer] R/W

Origin of feature: Driver		
Display name: Heartbeat Interval		
Range:[500–5000]	Default: ***3000***	Unit: ms

Upto Vimba V1.2.1

The driver sends a heartbeat request packet to the camera every `GVCPHBInterval` milliseconds. If the camera fails to respond to the heartbeat request, a retry is sent "`GVCPCmdTimeout`" ms later. After "`GVCPCmdRetries`" retries with no response, a camera unplugged event is returned by the driver.

This parameter can be increased significantly to bypass problems when debugging applications.

GevSCPSPacketSize [Integer] R/W

Origin of feature: Camera		
Range: [Camera dependent]	Default: ***Camera dependent***	Unit: Bytes

This parameter determines the Ethernet packet size. Generally, this number should be set to as large as the network card (or other involved active networking components) will allow. If this number is reduced, then CPU loading will increase. These large packet sizes (>1500) are called jumbo packets/frames in Ethernet terminology. If your GigE network card does not support jumbo packets/frames of at least 8228 bytes (the camera default on power-up), then you will need to reduce `GevSCPSPacketSize` parameter of the camera to match the maximum jumbo packet size supported by your GigE interface. A `GevSCPSPacketSize` of 1500 is a safe setting which all GigE network cards support.

If you are seeing all black images, or all frames reported as `StatFrameDropped` and zero images reported as `StatFrameDelivered`, you will likely need to decrease this parameter.

NonImagePayloadSize [Integer] R

Origin of feature: Camera	
Unit: Bytes	

Maximum size of chunk data, not including the image chunk, in the image block payload. If `ChunkModeActive` = ***`False`*** then ***`NonImagePayloadSize`*** = ***`0`***.

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 Allied Vision cameras via the `PtpAcquisitionGateTime` control. On Allied Vision GigE cameras, the device clock is represented by the camera `GevTimeStampValue` feature.

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

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

PtpAcquisitionGateTime [Integer] R/W

Origin of feature: Camera		
Range: [0 – (2^{63}-1)]	Default: ***0***	Unit: ns

`PtpAcquisition` trigger time. Used to schedule a synchronized software trigger on multiple PTP synchronized device. `PtpAcquisitionGateTime` must be set beyond current camera `GevTimeStampValue`, i.e., `GevTimeStampValue` ≥ `PtpAcquisitionGateTime`. When set below `GevTimeStampValue`, image acquisition stalls. `PtpAcquisitionGateTime` resets to zero when `PtpMode` set to ***Off***.

PtpMode [Enum] R/W

Origin of feature: Camera

Controls the PTP device behavior.

If using the camera event channel, a `EventPtpSyncLost` is sent if `PtpMode` is changed. `EventPtpSyncLocked` is sent once PTP synchronization is re-established.

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

Origin of feature: Camera

State of the PTP operation.

Disabled	[Default] Device **PtpMode** is set to ***Off***
Initializing	PTP is being initialized. If 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

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

PayloadSize [Integer] R

Origin of feature: Camera

Total size of payload, in bytes.

- If **ChunkModeActive** = ***True:***
  PayloadSize = **ImageSize** + **NonImagePayloadSize** + **8**
- If **ChunkModeActive** = ***False:***
  PayloadSize = **ImageSize**

Persistent

GevPersistentDefaultGateway [Integer] R/W

Origin of feature: Camera

Display name: Persistent Default Gateway

Persistent default gateway of the device.

GevPersistentIPAddress [Integer] R/W

Origin of feature: Camera
Display name: Persistent IP Address

Persistent IP address of the device.

GevPersistentSubnetMask [Integer] R/W

Origin of feature: Camera
Display name: Persistent Subnet Mask

Persistent subnet mask of the device.

StreamBytesPerSecond [Integer] R/W

Origin of feature: Camera	
Range: [1,000,000 – 124,000,000 (248,000,000 for GX in LAG mode)]	Unit: 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 Mb/s), or wireless networks. It is also an important control for multiple-camera situations. When multiple cameras are connected to a single GigE 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 Bytes/s is the typical maximum data rate for a GigE port. Beyond this setting, some network cards will drop packets.

If you are seeing occasional frames/packets reported as `StatFrameDropped`/`StatPacketMissed` you will likely need to decrease this parameter.

StreamFrameRateConstrain [Boolean] R/W

Origin of feature: Camera	
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 is not limited to bandwidth – only sensor readout time. Latter case is useful for `AcquisitionMode` = `Recorder` or `StreamHoldEnable = On` modes, as these modes are not bandwidth limited.

StreamHold

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 [Integer] R

Origin of feature: Camera
Unit: Frames

The maximum number of images (for the current size and format), which can be stored on the camera when `StreamHold` is enabled. Used when `AcquisitionMode` = ***Recorder***, or `StreamHoldEnable` = ***On***. This value is different for each camera depending on the camera internal memory size and the `ImageSize`.

StreamHoldEnable [Enum] R/W

Origin of feature: Camera

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

Allied Vision GigE cameras have a very accurate `timestamp` function for timestamping images.

Use PTP for synchronizing cameras.

GevTimestampControlLatch [Command]

Origin of feature: Camera

Captures timestamp and stores in `GevTimestampValue`.

GevTimestampControlReset [Command]

Origin of feature: Camera

Resets the camera's timestamp to 0. Not possible while PTP enabled (`PtpMode` = ***`Master`***, or ***`Auto`***).

GevTimestampTickFrequency [Integer] R

Origin of feature: Camera		
Range: [0–4294967295]	Default: ***`Camera dependent`***	Unit: Hz

Frequency of image timestamp. The image timestamp can be useful for determining whether images are missing from a sequence due to missing trigger events. Cameras offering clock synchronization via PTP will have a `GevTimeStampTickFrequency` of 1,000,000,000.

GevTimestampValue [Integer] R

Origin of feature: Camera
Unit: Camera clock ticks

Value of timestamp, when latched by `GevTimestampControlLatch`.

IO

The control and readout of all camera inputs and outputs. The number of inputs and outputs is camera model dependent.

StatusLED

StatusLedLevels [Integer] R/W

Origin of feature: Camera	
Range: [0–4294967296]	Default: ***`0`***

Status led levels in GPO mode.

`StatusLedPolarity` can invert these values.

StatusLedPolarity [Enum] R/W

Origin of feature: Camera
Possible values: ***`Normal`***, ***`Invert`***

Polarity applied to the status led specified by `StatusLedSelector`.

StatusLedSelector [Enum] R/W

Origin of feature: Camera

Possible values: ***StatusLed1***

Select the status led to be controlled with `StatusLedSource` and `StatusLedPolarity`.

StatusLedSource [Enum] R/W

Origin of feature: Camera

Signal source of the status led specified by `StatusLedSelector`.

GPO	General purpose output
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
LineIn1/2/3/4	External input ***Line1/2/3/4***
CCDTemperatureOK	Only for cameras that support this feature: indicates if camera has reached the desired temperature value
Strobe1	Source is strobe timing unit

Strobe

Definition

Strobe is an internal signal generator for on-camera clocking functions. *`Valid when any of the SyncOutSource is set to Strobe1`*. Strobe allows the added functionality of duration and delay, useful when trying to sync a camera exposure to an external strobe.

StrobeDelay [Integer] R/W

Origin of feature: Camera		
Range: [0 – Camera dependent]	Default: *`0`*	Unit: µs

Delay from strobe trigger to strobe output.

StrobeDuration [Integer] R/W

Origin of feature: Camera		
Range: [0 – Camera dependent]	Default: *`0`*	Unit: µs

Duration of strobe signal.

StrobeDurationMode [Enum] R/W

Origin of feature: Camera

Mode of the strobe timing unit.

`Source`	[Default] Strobe duration is the same as source duration
`Controlled`	Strobe duration is set by `StrobeDuration`

StrobeSource [Enum] R/W

Origin of feature: Camera

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

`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 the logic trigger signal inside of the camera. It is initiated by an external trigger or software trigger
`Exposing`	Active for the duration of sensor exposure
`FrameReadout`	Active for the duration of frame readout, i.e. the transferring of image data from the CCD to camera memory
`Acquiring`	Active during the acquisition stream

LineIn1	Active when there is an external trigger at line1
LineIn2	Active when there is an external trigger at line2
LineIn3	Active when there is an external trigger at line3
LineIn4	Active when there is an external trigger at line4

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

SyncIn

Signal source of the strobe timing unit. See `SyncOutSource` for descriptions.

SyncInGlitchFilter [Integer] R/W

Origin of feature: Camera		
Range: [0–50000]	Default: *0*	Unit: ns

Ignores glitches on the `SyncIn` input line with pulse duration less than set value.

Setting `SyncInGlitchFilter` value increases latency of *FrameTrigger* by same amount.

SyncInLevels [Integer] R

Origin of feature: Camera

A 4-bit register where each bit corresponds to a specific `SyncIn` input. For example, when this value returns *2* (0010), `SyncIn2` is high and all other Sync input signals are low.

SyncInSelector [Enum] R/W

Origin of feature: Camera	
Possible values: *SyncIn1*, *SyncIn2*, *SyncIn3*, *SyncIn4*	Default: *SyncIn1*

Select the sync-in line to control with `SyncInGlitchFilter`.

SyncOut

Used for synchronization with other cameras/devices or general purpose outputs.

SyncOutLevels [Integer] R/W

Origin of feature: Camera

Output levels of hardware sync outputs, for output(s) in ***GPO*** mode.

`SyncOutPolarity` can invert the `SyncOutLevels`.

SyncOutPolarity [Enum] R/W

Origin of feature: Camera	
Possible values: ***Normal***, ***Invert***	Default: ***Normal***

Polarity applied to the sync-out line specified by `SyncOutSelector`.

SyncOutSelector [Enum] R/W

Origin of feature: Camera	
Possible values: ***SyncOut1***, ***SyncOut2***, ***SyncOut3***, ***SyncOut4***	Default: ***SyncOut1***

Selects the sync-out line to control with `SyncOutSource`, `SyncOutPolarity`.

SyncOutSource [Enum] R/W

Origin of feature: Camera

Signal source of the sync-out line specified by `SyncOutSelector`.

GPO	General purpose output
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 when acquisition start has been initiated
LineIn1	Active when there is an external trigger at Line1
LineIn2	Active when there is an external trigger at Line2
LineIn3	Active when there is an external trigger at Line3
LineIn4	Active when there is an external trigger at Line4
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

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

ImageFormat

Height [Integer] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: Pixels

Height of image.

HeightMax [Integer] R

Origin of feature: Camera

Maximum image height for the current image mode.

ImageSize [Integer] R

Origin of feature: Camera

Size of images, in bytes, for the current format and size.

OffsetX [Integer] R/W

Origin of feature: Camera		
Range: [Camera dependent]	Default: *0*	Unit: Pixels

Starting column of the readout region (relative to the first column of the sensor) in pixels.

OffsetY [Integer] R/W

Origin of feature: Camera		
Range: [Camera dependent]	Default: *0*	Unit: Pixels

Starting row of the readout region (relative to the first row of the sensor) in pixels.

PixelFormat [Enum] R/W

Origin of feature: Camera

There are various pixel data formats that GigE cameras can output. Not all cameras have every mode (see the Technical Manuals for details).

Mono8	Bit depth: 8. One pixel every byte. Monochrome. For color cameras with on-camera interpolation, luminance (Y) channel returned.
Mono10	Bit depth: 10. One pixel every two bytes, LSB aligned. Monochrome. For color cameras with on-camera interpolation, luminance (Y) channel returned.
Mono12	Bit depth: 12. One pixel every two bytes, LSB aligned. Monochrome. For color cameras with on-camera interpolation, luminance (Y) channel returned.
Mono12Packed	Bit depth: 12. 2 pixels of data every 3 bytes. Monochrome. Doesn't support odd Width x Height.
Mono14	Bit depth: 14. One pixel every two bytes, LSB aligned. Monochrome. For color cameras with on-camera interpolation, luminance (Y) channel returned.
BayerGB8	Bit depth: 8. Raw, un-interpolated color. Interpolation performed by host software.
BayerRG8	Bit depth: 8. Raw, un-interpolated color. Interpolation performed by host software.
BayerGR8	Bit depth: 8. Raw, un-interpolated color. Interpolation performed by host software.
BayerBG8	Bit depth: 8. Raw, un-interpolated color. Interpolation performed by host software.
BayerBG10	Bit depth: 10. One pixel every two bytes, LSB aligned. Raw, un-interpolated color. Interpolation performed by host software.

BayerGB12Packed	Bit depth: 12. 2 pixels of data every 3 bytes. Raw, un-interpolated color. Interpolation performed by host software. Doesn't support odd Width or Height.
BayerGR12Packed	Bit depth: 12. 2 pixels of data every 3 bytes. Raw, un-interpolated color. Interpolation performed by host software. Doesn't support odd Width or Height.
BayerGB12	Bit depth: 12. One pixel every two bytes, LSB aligned. Raw, un-interpolated color. Interpolation performed by host software.
BayerRG12	Bit depth: 12. One pixel every two bytes, LSB aligned. Raw, un-interpolated color. Interpolation performed by host software.
BayerGR12	Bit depth: 12. One pixel every two bytes, LSB aligned. Raw, un-interpolated color. Interpolation performed by host software.
RGB8Packed	Bit depth: 8. One pixel every three bytes. On-camera interpolated color.
BGR8Packed	Bit depth: 8. One pixel every three bytes. On-camera interpolated color.
RGBA8Packed	Bit depth: 8. One pixel every four bytes. On-camera interpolated color. Alpha channel (A) is fully opaque, 0xFF
BGRA8Packed	Bit depth: 8. One pixel every four bytes. On-camera interpolated color. Alpha channel (A) is fully opaque, 0xFF
RGB12Packed	Bit depth: 12. One pixel every six bytes—R, G, B channels LSB-aligned. On-camera interpolated color.
YUV411Packed	Bit depth: 8. 4 pixel every 6 byte. On-camera interpolated color. Data in YUV411 format.
YUV422Packed	Bit depth: 8. 3 pixel every 6 byte. On-camera interpolated color. Data in YUV422 format.
YUV444Packed	Bit depth: 8. 2 pixel every 6 byte. On-camera interpolated color. Data in YUV444 format.

Width [Integer] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: Pixels

Width of image, in pixels.

WidthMax [Integer] R

Origin of feature: Camera

Maximum image width for the current image mode. Horizontal binning, for example, will change this value.

ImageMode

BinningHorizontal [Integer] R/W

Origin of feature: Camera	
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 ROI. Image sensitivity is also improved due to summed pixel charge / gray value.

- `BinningHorizontal` and `DecimationHorizontal` are mutually exclusive. Setting `BinningHorizontal` > 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.

BinningHorizontalMode [Enum] R/W

Origin of feature: Camera

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)

BinningVertical [Integer] R/W

Origin of feature: Camera	
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 ROI. Image sensitivity is also improved due to summed pixel charge / gray value.

- `BinningVertical` and `DecimationVertical` are mutually exclusive. Setting `BinningVertical` > 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.

BinningVerticalMode [Enum] R/W

Origin of feature: Camera

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)

DecimationHorizontal [Integer] R/W

Origin of feature: Camera	
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

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

DecimationVertical [Integer] R/W

Origin of feature: Camera	
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

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

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

Origin of feature: Camera	
Possible values: ***True***, ***False***	Default: ***False***

Flips the image sent by device horizontally. The ROI is applied after flipping.

ReverseY [Boolean] R/W

Origin of feature: Camera	
Possible values: ***True***, ***False***	Default: ***False***

Flips the image sent by device vertically. The ROI is applied after flipping.

SensorHeight [Integer] R/C

Origin of feature: Camera

The total number of pixel rows on the sensor.

SensorWidth [Integer] R/C

Origin of feature: Camera

The total number of pixel columns on the sensor.

Info

GevDeviceMACAddress [Integer] R/C

Origin of feature: Driver

Display name: Device MAC address

48-bit MAC address of the GVCP interface of the selected remote device.

DeviceFirmwareVersion [String] R/C

Origin of feature: Camera

Firmware version of this Allied Vision GigE camera.

DeviceID [String] R/C

Origin of feature: Camera

Serial number of the camera.

DeviceUserID [String] R/W

Origin of feature: Camera

Used for multiple-camera situations for providing meaningful labels to individual cameras.

DeviceModelName [String] R/C

Origin of feature: Camera

Camera model name, such as ***Manta G-125C***. Software should use the `DevicePartNumber` to distinguish between models.

DevicePartNumber [String] R/C

Origin of feature: Camera

Manufacturer's part number.

DeviceScanType [Enum] R/C

Origin of feature: Camera

Scan type of the camera.

DeviceVendorName [String] R/C

Origin of feature: Camera

Manufacturer's name: `Allied Vision Technologies.`

FirmwareVerBuild [Integer] R/C

Origin of feature: Camera

Build information.

FirmwareVerMajor [Integer] R/C

Origin of feature: Camera

Major part of the firmware version number (part before the decimal).

FirmwareVerMinor [Integer] R/C

Origin of feature: Camera

Minor part of firmware version number (part after the decimal).

SensorBits [Integer] R/C

Origin of feature: Camera

Maximum bit depth of sensor.

SensorType [Enum] R/C

Origin of feature: Camera

Type of image sensor. Monochrome or Bayer-pattern color sensor type.

SavedUserSets

Allied Vision 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.

LUT features cannot be saved.

To save the content of a LUT, use `Controls/LUTControl/LUTSave` or `LUTSaveAll`.

UserSetDefaultSelector [Enum] R/W

Origin of feature: Camera	
Possible values:	`Default`, `UserSet1`, `UserSet2`, `UserSet3`, `UserSet4`, `UserSet5`

On power-up or reset, this user set is loaded.

UserSetLoad [Command]

Origin of feature: Camera

Loads camera parameters from the user set specified by `UserSetSelector`.

UserSetSave [Command]

Origin of feature: Camera

Saves camera parameters to the user set specified by `UserSetSelector`. The `Default` setting cannot be overwritten.

UserSetSelector [Enum] R/W

Origin of feature: Camera	
Possible values:	`Default`, `UserSet1`, `UserSet2`, `UserSet3`, `UserSet4`, `UserSet5`

Selects a user set, for loading or saving camera parameters.

Stream

Info

GVSPFilterVersion [String] R/C

Origin of feature: Driver
Display name: GVSP Filter Version

Version of the GVSP Filter driver.

Multicast

Multicast mode allows the camera to send image data to all hosts on the same subnet as the camera. The host computer (or Vimba Viewer application instance) 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.

Most GigE switches support a maximum `PacketSize` of 1500 in multicast mode.

When using clients with Linux, you have to configure the IP subsystem to process Multicast IP traffic.

MulticastEnable [Boolean] R/W

Origin of feature: Driver	
Display name: Multicast Enable	
Possible values: *`True`*, *`False`*	Default: *`False`*

Enables multicast mode. In multicast mode all computers on the same subnet as the camera can receive image data from the camera `MulticastIPAddress`.

MulticastIPAddress [Integer] R/C

Origin of feature: Driver
Display name: Multicast IP Address

Sets the multicast IP address.

Settings

Definition GVSP = GigE Vision Streaming Protocol

GVSPAdjustPacketSize [Command]

Origin of feature: Driver
Display name: GVSP Adjust Packet Size

Requests the packet size used to be adjusted automatically.

GVSPBurstSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Burst Size		
Range: [1–256]	Default: ***32***	Unit: GVSP Packets

Maximum number of GVSP packets to be processed in a burst.

GVSPDriverSelector [Enum] R/W

Origin of feature: Driver	
Display name: GVSP Driver Selector	
Possible values: ***Filter***, ***Socket***	Default: ***Filter***

Streaming driver to be used.

GVSPHostReceiveBuffers [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Host Receive Buffers	
Range: [256–2048]	Default: ***512***

Number of buffers to be used by the network socket. Only applicable when not using the filter driver.

GVSPMaxLookBack [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Max Look Back	
Range: [1–1024]	Default: ***30***

Size of the look back window, in packets, when determining if a stream packet is missing. When a stream packet arrives out of order, the driver skips back `GVSPMaxLookBack` 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.

GVSPMaxRequests [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Max Requests	
Range: [1–512]	Default: ***3***

The maximum number of resend requests that the host will attempt before marking a packet dropped.

GVSPMaxWaitSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Max Wait Size		
Range: [8–1024]	Default: ***100***	

Maximum number of received GVSP packets following a resend request to wait before requesting again.

GVSPMissingSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Missing Size		
Range: [0–1024]	Default: ***512***	

Maximum number of simultaneous missing GVSP packets before dropping the frame (0 = OFF).

GVSPPacketSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Packet Size		
Range: [Camera dependent]	Default: ***Camera dependent***	Unit: Bytes

GVSP packet size.

GVSPTiltingSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Tilting Size		
Range: [0–1024]	Default: ***100***	

Maximum number GVSP packets received from a following frame before dropping the frame (0 = OFF).

GVSPTimeout [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Timeout		
Range: [10–5000]	Default: ***70***	Unit: ms

End of stream timeout. If no stream packet received before `GVSPTimeout`, host requests resend, up to `GVSPMaxRequests` times. If still no packet received from camera, packet is marked as dropped.

Statistics

The packet counts in these statistics cover the image transport. Packets used for camera control or event data are not counted. All counters are reset at `AcquisitionStart`.

StatFrameRate [Float] R

Origin of feature: Driver
Display name: Stat Frame Rate

Rate at which the device is acquiring frames, derived from the frame timestamps.

StatFrameDelivered [Integer] R

Origin of feature: Driver
Display name: Stat Frames Delivered

Number of error-free frames captured since the start of imaging.

StatFrameDropped [Integer] R

Origin of feature: Driver
Display name: Stat Frames Dropped

Number of incomplete frames received by the host due to missing packets (not including shoved frames).

StatFrameRescued [Integer] R

Origin of feature: Driver
Display name: Stat Frames Rescued

Number of frames that initially had missing packets but were successfully completed after packet resend.

StatFrameShoved [Integer] R

Origin of feature: Driver
Display name: Stat Frames Shoved

Number of frames dropped because the transfer of a following frame was completed earlier.

StatFrameUnderrun [Integer] R

Origin of feature: Driver
Display name: Stat Frames Underrun

Number of frames missed due to the non-availability of a user supplied buffer.

StatLocalRate [Float] R

Origin of feature: Driver
Display name: Stat Local Rate

Inverse of time interval between the last two frames (faulty or not) received by the host. No averaging is performed.

In case of error-free frame reception, `StatLocalRate` is similar to `StatFrameRate`, except that the host clock is used instead of frame timestamps for measuring the time interval between frames. Otherwise, `StatLocalRate` and `StatFrameRate` may differ significantly.

StatPacketErrors [Integer] R

Origin of feature: Driver
Display name: Stat Packets Errors

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

StatPacketMissed [Integer] R

Origin of feature: Driver
Display name: Stat Packets Missed

Number of packets missed since the start of imaging.

If everything is configured correctly, this number should remain zero, or at least very low compared to `StatPacketReceived`.

StatPacketReceived [Integer] R

Origin of feature: Driver
Display name: Stat Packets Received

Number of error-free packets received by the driver since the start of imaging, this number should grow steadily during continuous acquisition.

StatPacketRequested [Integer] R

Origin of feature: Driver
Display name: Stat Packets Requested

Number of missing packets that were requested to be resent from the device.

If everything is configured correctly, this number should remain zero, or at least very low compared to `StatPacketReceived`.

StatPacketResent [Integer] R

Origin of feature: Driver
Display name: Stat Packets Resent

Number of packets resent by the camera since the start of imaging.

StatTimeElapsed [Float] R

Origin of feature: Driver
Display name: Stat Time Elapsed

Elapsed time (in seconds) since the streaming was started.

StreamInformation

StreamID [String] R/C

Origin of feature: Driver
Display name: Stream ID

Vimba V1.3 or later Device's unique ID for the stream.

StreamType [Enum] R/C

Origin of feature: Driver
Display name: Stream Type

Vimba V1.3 or later Identifies the transport layer technology of the stream.

GigE IR & scientific camera and driver features

Goldeye G and Goldeye G Cool are short-wave infrared GigE cameras that provide comprehensive and advanced set of features for infrared imaging requirements.

This chapter describes the standard and advanced camera controls, as seen from the Vimba Viewer, for Goldeye G and Goldeye G Cool cameras using the GenICam standard feature naming convention.

AcquisitionControl

This group of controls relates to image acquisition.

AcquisitionAbort [Command]

Origin of feature: Camera

Software command to stop camera from receiving frame triggers and abort the current acquisition. A partially transferred image will be completed.

AcquisitionFrameCount [Integer] R/W

Origin of feature: Camera		
Range:[1–65535]	Default: ***1***	Unit: Frames

Defines 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`.

AcquisitionFrameRate [Float] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: Frames per second

When `TriggerSelector` = ***`FrameStart`*** and either `TriggerMode` = ***`Off`*** or `TriggerSource` = ***`FixedRate`***, this control specifies the frame rate. Depending on the exposure duration, the camera may not achieve the frame rate set here.

AcquisitionFrameRateLimit [Float] R

Origin of feature: Camera	
Range: [Camera dependent]	Unit: Frames per second

The maximum frame rate possible for the current exposure duration and image format.

AcquisitionMode [Enum] R/W

Origin of feature: Camera

Determines the behavior of the camera when acquisition start is triggered.

Continuous	[Default] After an acquisition start event, the camera will continuously receive frame trigger events. See `TriggerSelector` and `TriggerSource` for more information.
SingleFrame	The camera will only deliver a single frame trigger event. Further trigger events will be ignored until acquisition is stopped and restarted
MultiFrame	The camera will acquire the number of images specified by `AcquisitionFrameCount`. Further trigger events will be ignored until acquisition is stopped and restarted
Recorder	The camera will continuously record images into the camera on-board memory, but will not send them to the host until an ***AcquisitionRecord*** trigger signal is received. Further ***AcquisitionRecord*** trigger events will be ignored until acquisition is stopped and restarted. Combined with the `RecorderPreEventCount` control, this feature is useful for returning any number of frames before a trigger event. When ***AcquisitionRecord*** trigger is received, the currently imaging/acquiring image will complete as normal, and then at least one more image will be taken. The memory is a circular buffer, that starts rewriting images once it is full. Its size is determined by `AcquisitionFrameCount`

AcquisitionStart [Command]

Origin of feature: Camera

Software command to start camera receiving frame triggers. Valid when `TriggerMode` = ***Off***. See `TriggerSelector` = ***FrameStart*** trigger.

AcquisitionStop [Command]

Origin of feature: Camera

Software command to stop camera from receiving frame triggers. Valid when `TriggerMode` = ***Off***. See `TriggerSelector` = ***FrameStart*** trigger.

ExposureMode [Enum] R/W

Origin of feature: Camera	
Timed	[Default] Camera exposure time is set by `ExposureTime`

Control for exposure duration.

ExposureTime [Float] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: µs

The sensor integration time.

IntegrationMode [Enum] R/W

Origin of feature: Camera	
`IntegrateThenRead`	[Default] The integration interval is not allowed to overlap with the readout
`IntegrateWhileRead`	The integration interval is allowed to overlap with the readout

RecorderPreEventCount [Integer] R/W

Origin of feature: Camera		
Range:[0–65535]	Default: ***`0`***	Unit: Frames

Valid when `AcquisitionMode` = ***`Recorder`***. The number of frames returned before the ***`AcquisitionRecord`*** trigger event, with `AcquisitionFrameCount minus RecorderPreEventCount` frames being returned after the ***`AcquisitionRecord`*** trigger event. `RecorderPreEventCount` must be less than or equal to `AcquisitionFrameCount`.

At least one image must be captured after the ***`AcquisitionRecord`*** trigger event, i.e., you cannot set `RecorderPreEventCount` = ***`1`***, and `AcquisitionFrameCount` = ***`1`***.

TriggerActivation [Enum] R/W

Origin of feature: Camera

Type of activation, for hardware triggers. This controls edge/level and polarity sensitivities.

RisingEdge	[Default] Rising edge trigger
FallingEdge	Falling edge trigger
AnyEdge	Rising or falling edge
LevelHigh	Active high signal
LevelLow	Active low signal

TriggerDelay [Float] R/W

Origin of feature: Camera		
Range:[0 – Camera dependent]	Default: ***0***	Unit: µs

Start-of-image can be delayed to begin some time after a trigger event is received by the camera. This feature is valid only when `TriggerSource` is set to external trigger (i.e. ***Line1***, ***Line2***). This control is a common trigger to sync with a strobe lighting source, which will inherently have some fixed setup time.

TriggerMode [Enum] R/W

Origin of feature: Camera

Enables or disables trigger set in `TriggerSelector`.

Off	Trigger disabled
On	[Default] Trigger enabled

If `TriggerMode` = ***Off*** and `TriggerSelector` = ***FrameStart***, images triggered in ***FixedRate*** at `AcquisitionFrameRateAbs`.

TriggerOverlap [Enum] R/W

Origin of feature: Camera

Permitted window of trigger activation, relative to previous frame. Does not work with Software triggering. Only external.

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

TriggerSelector [Enum] R/W

Origin of feature: Camera

Selects a trigger, then use the controls {`TriggerMode`, `TriggerSoftware`, `TriggerSource`, `TriggerActivation`, `TriggerOverlap`, `TriggerDelay`} to setup and read the trigger features.

FrameStart	[Default] The trigger which starts each image (when acquisition is running)
AcquisitionStart	The trigger which starts the acquisition process
AcquisitionEnd	The trigger which ends the acquisition process
AcquisitionRecord	The trigger which initiates the sending of `AcquisitionFrameCount` number of recorded images from the camera on-board memory to the host

TriggerSoftware [Command]

Origin of feature: Camera

Triggers an image. Valid when `TriggerSource` = ***Software***.

TriggerSource [Enum] R/W

Origin of feature: Camera

Determines how an image frame is initiated within an acquisition stream.

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

Freerun	[Default] Camera runs at maximum supported frame rate depending on the exposure time and ROI size
Line1	External trigger ***Line1***
Line2	External trigger ***Line2***
FixedRate	Camera self-triggers at a fixed frame rate defined by `AcquisitionFrameRate`
Software	Software initiated image capture

AnalogControl

SensorGain [Enum] R/W

Origin of feature: Camera

Sets the FPA gain level.

`Gain0`	[default] Sets FPA gain to lowest level
`Gain1`	Sets FPA gain to a higher level than `Gain0` (if available)
`Gain2`	Sets FPA gain to a higher level than `Gain1` (if available)

BufferHandlingControl

StreamAnnounceBufferMinimum [Integer] R/C

Origin of feature: Driver

Display name: Stream Announce Buffer Minimum

Vimba V1.3 or later Minimal number of buffers to announce to enable selected acquisition mode.

StreamAnnouncedBufferCount [Integer] R

Origin of feature: Driver

Display name: Stream Announced Buffer Count

Vimba V1.3 or later Number of announced (known) buffers on this stream.

StreamBufferHandlingMode [Enum] R/W

Origin of feature: Driver

Display name: Stream Buffer Handling Mode

Vimba V1.3 or later Available buffer handling modes of this stream.

ChunkDataControl

ChunkModeActive [Boolean] R/W

Origin of feature: Camera	
Possible values: *True*, *False*	Default: *False*

Enables camera to send GigE Vision Standard Protocol chunk data with an image. The table below presents currently implemented chunk data:

[Bytes 1 – 4] Acquisition count
[Byte 5 – 8] Reserved. 0
[Bytes 9 – 12] Exposure value in µs.
[Bytes 13 – 16] Gain value in 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.

`ChunkModeActive` is read only during acquisition.

NonImagePayloadSize [Integer] R

Origin of feature: Camera
Unit: Bytes

Maximum size of chunk data, not including the image chunk, in the image block payload. If `ChunkModeActive` = *`False`* then `NonImagePayloadSize` = *0*. If `ChunkModeActive` = *`True`* then `NonImagePayloadSize` = *48*.

DeviceControl

BandwidthControlMode [Enum] R/W

Origin of feature: Camera

Selects the desired mode of bandwidth control. Bandwidth allocation can be controlled by `DeviceLinkThroughputLimit` or by SCPD0 register. If you are not familiar with SCPD0 and how this driver uses this register, leave this feature set to `DeviceLinkThroughputLimit`.

`DeviceLinkThroughputLimit`	[Default] See the `DeviceLinkThroughputLimit` feature for more information
`SCPD`	Stream channel packet delay expressed in timestamp 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; for example, 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

DeviceFamilyName [String] R/C

Origin of feature: Camera

Identifier of the product family of the device.

DeviceFanMode [Enum] R/W

Origin of feature: Camera

Enables or disables the fan.

`On`	[Default] `Turns the`
`Off`	Turns the device fan off

DeviceFanRpm [Integer] R

Origin of feature: Camera

Range: [0 – Camera dependent]	Unit: rpm

Current rotation speed of the fan.

DeviceFanSelector [Enum] R/W

Origin of feature: Camera
Possible values: `Main`

Selects the fan to be controlled by `DeviceFanMode` and `DeviceFanRpm`.

DeviceFirmwareVersion [String] R/C

Origin of feature: Camera

Firmware version of the device.

DeviceLinkHeartbeatTimeout [Float] R/W

Origin of feature: Camera		
Granularity: 1000	Default: `3000000`	Unit: µs

Controls the current heartbeat timeout of the link selected by `DeviceLinkSelector`.

DeviceLinkSelector [Integer] R/W

Origin of feature: Camera

Selects which link of the device to control.

DeviceLinkThroughputLimit [Integer] R/W

Origin of feature: Camera		
Range: [1,000,000 – 124,000,000]	Default: `115,000,000`	Unit: 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 Mb/s), or wireless networks. It is also an important control for multiple-camera situations. When multiple cameras are connected to a single GigE port (usually through a switch), `DeviceLinkThroughputLimit` for each camera needs to be set to a value so that the sum of each camera's `DeviceLinkThroughputLimit` 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 `DeviceLinkThroughputLimit` setting for a camera in any image mode, use the following formula:

DeviceLinkThroughputLimit = Height x Width x FrameRate x Bytes per Pixel

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

If you are seeing occasional frames/packets reported as `StatFrameDropped`/ `StatPacketMissed` you will likely need to decrease this parameter.

DeviceLinkThroughputLimitMode [Boolean] R/W

Origin of feature: Camera	
Possible values: *On*, *Off*	Default: *On*

When *On*, camera automatically limits frame rate to bandwidth, determined by `DeviceLinkThroughputLimit`, to prevent camera buffer overflows and dropped frames. If *Off*, frame rate is not limited to bandwidth but by sensor readout time. Latter case is useful for `AcquisitionMode` = `Recorder` or `StreamHoldEnable = On` modes, as these modes are not bandwidth limited.

DeviceModelName [String] R/C

Origin of feature: Camera

Camera family and model name, such as *Goldeye G-032*. Software should use the `DevicePartNumber` to distinguish between models.

DeviceRelativeHumidity [Float] R

Origin of feature: Camera

Relative humidity, in percent, measured at the location selected in `DeviceRelativeHumiditySelector`.

DeviceRelativeHumiditySelector [Enum] R/W

Origin of feature: Camera
Possible values: *Sensorboard*

Selects the location for measuring relative humidity.

DeviceSFNCVersionMajor [Integer] R/C

Origin of feature: Camera

Major part of the SFNC version number (part before the decimal).

DeviceSFNCVersionMinor [Integer] R/C

Origin of feature: Camera

Minor part of the SFNC version number (part after the decimal).

DeviceSFNCVersionSubMinor [Integer] R/C

Origin of feature: Camera

Subordinate part of the firmware Minor number (part after the minor).

DeviceScanType [Enum] R/C

Origin of feature: Camera

Scan type of the camera: ***`Areascan`***.

DeviceSerialNumber [String] R/C

Serial number of the camera.

Origin of feature: Camera

DeviceStreamChannelPacketSize [Integer] R/W

Origin of feature: Camera		
Range: [0 – Camera dependent]	Default: ***`8999`***	Unit: Bytes

Specifies the stream packet size to send on the selected channel for the camera or specifies the maximum packet size supported by the receiver.

DeviceStreamChannelSelector [Integer] R/W

Origin of feature: Camera	
Range: [0 – Camera dependent]	Default: ***`0`***

Selects the stream channel to control.

DeviceTLType [Enum] R/C

Origin of feature: Camera

Defines the transport layer type: **`GigEVision`**.

DeviceTemperature [Float] R

Origin of feature: Camera

Device temperature, in °C, measured at the location selected by **`DeviceTemperatureSelector`**.

DeviceTemperatureSelector [Enum] R/W

Origin of feature: Camera	
Possible values: ***Sensor***, ***Sensorboard***, ***Mainboard***	Default: ***Sensor***

Selects the location of temperature measurement points within the camera.

DeviceType [Enum] R/C

Origin of feature: Camera

Type of the camera: ***Transmitter***.

DeviceUserID [String] R/W

Origin of feature: Camera

Used for multiple-camera situations for providing meaningful labels to individual cameras.

DeviceVendorName [String] R/C

Origin of feature: Camera

Manufacturer's name: ***Allied Vision Technologies***.

SensorCoolingPower [Float] R

Origin of feature: Camera

Cooling power consumption in mW.

SensorTemperatureControlMode [Enum] R/W

Origin of feature: Camera

Defines the control mode for the thermoelectric cooler (TEC) of the sensor.

Off	No sensor temperature control
TemperatureControl	[Default] Regulates the sensor temperature in accordance with active values of other SensorTemperature features below

SensorTemperatureControlState [Enum] R

Origin of feature: Camera

Displays the state of sensor temperature control.

Off	Sensor cooling is off
Deviated	Sensor temperature deviates from the setpoint value

Stable	Sensor temperature is stable at the setpoint
LowerLimit	Cooling regulator is working at its lower limit
UpperLimit	Cooling regulator is working at its upper limit
Alert	Camera temperature above threshold temperature

SensorTemperatureSetpointActivate [Command]

Origin of feature: Camera

Activates the currently selected `SensorTemperatureSetpointSelector`.

SensorTemperatureSetpointActive [Enum] R

Origin of feature: Camera

Possible values: *1*, *2*, *3*, *4*

Displays the active setpoint.

SensorTemperatureSetpointMode [Enum] R/W

Origin of feature: Camera

Controls the setpoint mode for the TEC.

Manual	`The setpoint has to be chosen`
Auto	[Default] The setpoint is chosen automatically

SensorTemperatureSetpointSelector [Enum] R/W

Origin of feature: Camera

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

Selects the setpoint to be activated.

SensorTemperatureSetpointValue [Float] R/W

Origin of feature: Camera

The setpoint temperature, in °C, corresponding to the setpoint selected in `SensorTemperatureSetpointSelector.`

TimestampLatch [Command]

Origin of feature: Camera

Captures timestamp and stores in `TimestampLatchValue`.

TimestampLatchValue [Integer] R

Origin of feature: Camera
Unit: Camera clock ticks

Value of timestamp, when latched by `TimestampLatch`.

TimestampReset [Command]

Origin of feature: Camera

Resets the camera's timestamp to 0.

DigitalIOControl

LineIn

Signal source of the strobe timing unit. See `LineOutSource` for descriptions.

LineInGlitchFilter [Integer] R/W

Origin of feature: Camera		
Range: [0–50000]	Default: ***0***	Unit: ns

Ignores glitches on the `LineIn` input line with pulse duration less than set value.

Setting `LineInGlitchFilter` value increases latency of ***`FrameTrigger`*** by same amount.

LineInLevels [Integer] R

Origin of feature: Camera

A 4-bit register where each bit corresponds to a specific `LineIn` input. For example, when this value returns *2* (0010), `LineIn2` is high and all other Line input signals are low.

LineInSelector [Enum] R/W

Origin of feature: Camera	
Possible values: ***`LineIn1`***, ***`LineIn2`***	Default: ***`LineIn1`***

Select the ***`LineIn`*** to control with `LineInGlitchFilter`.

LineOut

Used for synchronization with other cameras/devices or general purpose outputs.

LineOutLevels [Integer] R/W

Origin of feature: Camera

Output levels of hardware line outputs, for output(s) in ***GPO*** mode.

`LineOutPolarity` can invert the `LineOutLevels`.

LineOutPolarity [Enum] R/W

Origin of feature: Camera	
Possible values: ***Normal***, ***Invert***	Default: ***Normal***

Polarity applied to the ***LineOut*** specified by `LineOutSelector`.

LineOutSelector [Enum] R/W

Origin of feature: Camera	
Possible values: ***LineOut1***, ***LineOut2***	Default: ***LineOut1***

Selects the ***LineOut*** to control with `LineOutSource`, `LineOutPolarity`.

LineOutSource [Enum] R/W

Origin of feature: Camera

Signal source of the ***LineOut*** line specified by `LineOutSelector`.

GPO	General purpose output
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	[Default] 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 when acquisition start has been initiated

`LineIn1`	Active when there is an external trigger at `Line1`
`LineIn2`	Active when there is an external trigger at `Line2`
`Strobe1`	The output signal is controlled according to Strobe1 settings

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

Strobe

Definition

Strobe is an internal signal generator for on-camera clocking functions. Valid when any of the `LineOutSource` is set to `Strobe1`. Strobe allows the added functionality of duration and delay, which is useful when trying to sync a camera exposure to an external strobe.

StrobeDelay [Integer] R/W

Origin of feature: Camera		
Range: [0 – Camera dependent]	Default: `0`	Unit: µs

Delay from strobe trigger to strobe output.

StrobeDuration [Integer] R/W

Origin of feature: Camera		
Range: [0 – Camera dependent]	Default: `0`	Unit: µs

Duration of strobe signal.

StrobeDurationMode [Enum] R/W

Origin of feature: Camera

Mode of the strobe timing unit.

`Source`	[Default] Strobe duration is the same as source duration
`Controlled`	Strobe duration is set by `StrobeDuration`

StrobeSource [Enum] R/W

Origin of feature: Camera

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

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 the logic trigger signal inside of the camera. It is initiated by an external trigger or software trigger
Exposing	Active for the duration of sensor exposure
FrameReadout	Active for the duration of frame readout, i.e. the transferring of image data from the CCD to camera memory
Acquiring	Active during the acquisition stream
LineIn1	Active when there is an external trigger at line1
LineIn2	Active when there is an external trigger at line2

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

EventControl

EventData

Origin of all features: Camera

The following table lists all the events supported by the camera:

EventAcquisitionEndFrameID	*EventFrameTriggerReadyFrameID*
EventAcquisitionEndTimestamp	*EventFrameTriggerReadyTimestam*
EventAcquisitionRecordTriggerFrameID	*EventLine1FallingEdgeFrameID*
EventAcquisitionRecordTriggerTimestam	*EventLine1FallingEdgeTimestamp*
EventAcquisitionStartFrameID	*EventLine1RisingEdgeFrameID*
EventAcquisitionStartTimestamp	*EventLine1RisingEdgeTimestamp*
EventErrorFrameID	*EventLine2FallingEdgeFrameID*
EventErrorTimestamp	*EventLine2FallingEdgeTimestamp*
EventExposureEndFrameID	*EventLine2RisingEdgeFrameID*
EventExposureEndTimestamp	*EventLine2RisingEdgeTimestamp*
EventFrameTriggerFrameID	*EventOverflowFrameID*
EventFrameTriggerTimestamp	*EventOverflowTimestamp*

EventID

Origin of all features: Camera

EventAcquisitionStart [Integer] R/C	*40000*
EventAcquisitionEnd [Integer] R/C	*40001*
EventFrameTrigger [Integer] R/C	*40002*
EventExposureEnd [Integer] R/C	*40003*
EventAcquisitionRecordTrigger [Integer] R/C	*40004*
EventLine1RisingEdge [Integer] R/C	*40010*
EventLine1FallingEdge[Integer] R/C	*40011*
EventLine2RisingEdge [Integer] R/C	*40012*
EventLine2FallingEdge [Integer] R/C	*40013*
EventFrameTriggerReady [Integer] R/C	*40018*
EventOverflow [Integer] R/C	*65534*
EventError [Integer] R/C	*65535*

If you use the message channel for event notification, you are always subscribed to `EventOverflow` and `EventError` events.

There is no mechanism to detect the loss of events during transportation. If misconfigured, cameras may produce lots of events—more than a PC can handle.

EventNotification [Enum] R/W

Origin of feature: Camera	
Possible values: `On`, `Off`	Default: `Off`

Activates event notification on the GigE Vision message channel.

EventSelector [Enum] R/W

Origin of feature: Camera

Selects a specific event to be enabled or disabled using `EventNotification`. Possible values are listed as following:

`AcquisitionStart` [Default]	`AcquisitionEnd`
`FrameTrigger`	`ExposureEnd`
`AcquisitionRecordTrigger`	`Line1RisingEdge`
`Line1FallingEdge`	`Line2RisingEdge`
`Line2FallingEdge`	`FrameTriggerReady`

EventsEnable1 [Integer] R/W

Origin of feature: Camera
Default: `0`

Bit field of all events. For example:

`Bit 1`	`EventAcquisitionStart`
`Bit 2`	`EventAcquisitionEnd`
`Bit 3`	`EventFrameTrigger`
`Bit 19`	`EventFrameTriggerReady`

This is an alternative to setting each event individually using the `EventNotification` and `EventSelector` method.

FileAccessControl

FileAccessBuffer [Register] R/W

Origin of feature: Camera

Defines the intermediate access buffer that allows the exchange of data between the camera file storage and the application.

FileAccessLength [Integer] R/W

Origin of feature: Camera	
Range: ≥ 0	Unit: Bytes

Controls the length of mapping between the camera file storage and the `FileAccessBuffer`.

FileAccessOffset [Integer] R

Origin of feature: Camera		
Range: ≥ 0	Default: ***0***	Unit: Bytes

Controls the offset of mapping between the camera file storage and the `FileAccessBuffer`.

FileAttribute [Integer] R

Origin of feature: Camera

Attribute of the currently selected file.

Bit 0–1	These two bits are used to encode the privilege level for a file. It defines the owner of the file: • ***0*** = [Default] User owns the file. User can overwrite/delete the file • ***1*** = For factory personnel use only • ***2***, ***3*** = Reserved
Bit 2–31	Reserved, always 0

FileAttributeBuffer [Integer] R/(W)

Origin of feature: Camera

Contains the attribute that will be used for newly created files when `FileOperationSelector` = ***`WriteAttribute`***.

Bit 0–1	These two bits are used to encode the privilege level for a file. It defines the owner of the file: • ***0*** = [Default] User owns the file. User can overwrite/delete the file • ***1*** = For factory personnel use only • ***2***, ***3*** = Reserved
Bit 2–31	Reserved, always 0

FileDescription [String] R

Origin of feature: Camera

Description string for currently selected file. A maximum of 32 characters are allowed, including the trailing NULL character.

FileDescriptionBuffer [String] R/W

Origin of feature: Camera

Contains the description that will be used for newly created files when `FileOperationSelector` = ***`WriteDescription`***. A maximum of 32 characters are allowed, including the trailing NULL character.

FileOpenMode [Enum] R/W

Origin of feature: Camera

Selects the access mode in which a file is opened in the device.

`Read`	[Default] Selects read-only open mode
`Write`	Selects write-only open mode

FileOperationExecute [Command]

Origin of feature: Camera

Executes the operation selected by `FileOperationSelector` on the selected file.

FileOperationResult [Integer] R

Origin of feature: Camera

Unit: Bytes

Presents the result of the file operation. For read or write operations, the number of successfully read/written bytes is returned.

FileOperationSelector [Enum] R/W

Origin of feature: Camera

Selects the target operation for the selected file in the device. This operation is executed when the `FileOperationExecute` feature is called.

`Open`	[Default] Opens the file selected by `FileSelector` in the device with an access mode selected in `FileOpenMode`
`Close`	Closes the file selected by `FileSelector` in the device

Read	Reads "FileAccessLength" bytes from the device storage, at the file relative offset set in FileAccessOffset into FileAccessBuffer
Write	Writes "FileAccessLength" bytes taken from the FileAccessBuffer into the device storage at the file relative offset defined by FileAccessOffset
Delete	Deletes the file selected by FileSelector in the device. Note: Deleting a device file does not remove the associated FileSelector entry to allow future operation on this file
WriteType	Writes the FileType taken from the FileTypeBuffer into the device storage
WriteAttribute	Writes the FileAttribute taken from the *FileAttributeBuffer* into the device storage
WriteDescription	Writes the FileDescription taken from the FileDescriptionBuffer into the device storage

FileOperationStatus [Enum] R

Origin of feature: Camera

Shows the status of file operation execution.

Success	File operation successful
Failure	File operation failed

FileSelector [Enum] R/W

Origin of feature: Camera

Selects the target file in the device. The entries of this enumeration define the names of all files in the device that can be accessed via the file access. For example:

- DPC_000: Defect pixel correction dataset 0
- NUC_001: Non-uniformity correction dataset 1

FileSize [Integer] R

Origin of feature: Camera

Represents the size of the selected file in bytes.

FileStatus [Enum] R

Origin of feature: Camera

Possible values: *Closed*, *Open*

Presents the status of the file

FileType [Integer] R

Origin of feature: Camera

Type of currently selected file.

FileTypeBuffer [Integer] R/(W)

Origin of feature: Camera

Possible values:
0x1000 = Non-uniformity correction data
0x2000 = Defect pixel correction data

Contains the type that will be used for newly created files when `FileOperationSelector` = ***WriteType***.

GigE

Configuration

GevIPConfigurationApply [Command]

Origin of feature: Driver

Display name: IP Configuration Apply

Apply the IP configuration mode selected by `GevIPConfigurationMode`.

GevIPConfigurationMode [Enum] R/W

Origin of feature: Driver

Display name: IP Configuration Mode

Possible values: ***LLA***, ***DHCP***, ***Persistent***

Current IP configuration mode.

Current

GevCurrentDefaultGateway [Integer] R

Origin of feature: Driver

Display name: Current Default Gateway

IP address of the default Gateway of the device.

GevCurrentIPAddress [Integer] R

Origin of feature: Driver
Display name: Current IP Address

Current IP address of the device.

GevCurrentSubnetMask [Integer] R

Origin of feature: Driver
Display name: Current Subnet Mask

Current Subnet Mask of the device.

GVCP

Definition

GVCP = GigE Vision Control Protocol
Allied Vision GigE IR and scientific cameras have a sophisticated real time resend mechanism that ensures a high degree of data integrity.

GVCPCmdRetries [Integer] R/W

Origin of feature: Driver	
Display name: Command Retries	
Range:[1–9]	Default: ***5***

Controls the maximum number of resend requests that the host will attempt when trying to recover a lost packet.

GVCPCmdTimeout [Integer] R/W

Origin of feature: Driver		
Display name: Command Timeout		
Range:[100–1000]	Default: ***250***	Unit: ms

Timeout waiting for an answer from the device.

GevHeartbeatInterval [Integer] R/W

Origin of feature: Driver		
Display name: Heartbeat Interval		
Range:[200–1450]	Default: ***1450***	Unit: ms

Vimba V1.3 or later

The driver sends heartbeat packets to the camera every `GevHeartbeatInterval` milliseconds. If the camera fails to respond to the heartbeat request, a retry is sent "`GVCPCmdTimeout`" ms later. After "`GVCPCmdRetries`" retries with no response, a camera unplugged event is returned by the driver.

This parameter can be increased significantly to bypass problems when debugging applications.

GevSCPSPacketSize [Integer] R/W

Origin of feature: Camera		
Range: [Camera dependent]	Default: ***Camera dependent***	Unit: Bytes

This parameter determines the Ethernet packet size. Generally, this number should be set to as large as the network card (or other involved active networking components) will allow. If this number is reduced, then CPU loading will increase. These large packet sizes (>1500) are called jumbo packets/frames in Ethernet terminology. If your GigE network card does not support jumbo packets/frames of at least 8228 bytes (the camera default on power-up), then you will need to reduce `GevSCPSPacketSize` parameter of the camera to match the maximum jumbo packet size supported by your GigE interface. A `GevSCPSPacketSize` of 1500 is a safe setting which all GigE network cards support.

If you are seeing all black images, or all frames reported as `StatFrameDropped` and zero images reported as `StatFrameDelivered`, you will likely need to decrease this parameter.

Persistent

GevPersistentDefaultGateway [Integer] R/W

Origin of feature: Camera
Display name: Persistent Default Gateway

Persistent default gateway of the device.

GevPersistentIPAddress [Integer] R/W

Origin of feature: Camera
Display name: Persistent IP Address

Persistent IP address of the device.

GevPersistentSubnetMask [Integer] R/W

Origin of feature: Camera
Display name: Persistent Subnet Mask

Persistent subnet mask of the device.

ImageCorrectionControl

BackgroundCorrection

BCDatasetMeanValue [Integer] R

Origin of feature: Camera

Provides mean value of the correction image.

BCDatasetOffsetValue [Integer] R/W

Origin of feature: Camera	
Range: [-32768 to 32768]	Default: ***0***

Specifies the output offset of the corrected image. The scale is always based on the maximum pixel depth the camera supports, independent of the active output pixel format.

BCDatasetROIHeight [Integer] R

Origin of feature: Camera

ROI height of the integrated correction image. Background correction stays active as long as the effective ROI of the camera fully fits into the correction image's ROI. See `BCState` feature for current state of the background correction processing.

BCDatasetROIOffsetX [Integer] R

Origin of feature: Camera

ROI horizontal offset of the integrated correction image. Background correction stays active as long as the effective ROI of the camera fully fits into the correction image's ROI. See `BCState` feature for current state of the background correction processing.

BCDatasetROIOffsetY [Integer] R

Origin of feature: Camera

ROI vertical offset of the integrated correction image. Background correction stays active as long as the effective ROI of the camera fully fits into the correction image's ROI. See `BCState` feature for current state of the background correction processing.

BCDatasetROIWidth [Integer] R

Origin of feature: Camera

ROI width of the integrated correction image. Background correction stays active as long as the effective ROI of the camera fully fits into the correction image's ROI. See `BCState` feature for current state of the background correction processing.

BCIntegrationAbort [Command]

Origin of feature: Camera

Aborts a running integration as soon as possible. The correction buffer will be invalid if lesser number of frames have been integrated than requested.

BCIntegrationFrameCount [Integer] R/W

Origin of feature: Camera	
Range: [1 – 4]	Default: ***1***

Number of frames to integrate after `BCIntegrationStart` command. Integrating more images improves the correction quality because influence of dynamic noise on the correction image is reduced.
`BCIntegrationFrameCount` is always rounded off to the next power of two.

BCIntegrationMode [Enum] R/W

Origin of feature: Camera

Controls how a background correction image will be acquired upon `BCIntegrationStart` command.

Integrate	[Default] After `BCIntegrationStart`, a correction image that is the mean of `BCIntegrationFrameCount` images will be acquired
FrameBuffer	Stores every frame to the correction memory and uses the previously stored image for correction. If `BCMode` = ***On***, this can be used to get a dynamic frame-to-frame difference of the live image. Use `BCIntegrationStart` to start the ***FrameBuffer*** writing, set `BCIntegrationMode` = ***Integrate*** to stop it

BCIntegrationStart [Command]

Origin of feature: Camera

Starts the integration of `BCIntegrationFrameCount` frames, depending on `BCIntegrationMode`. This command does not control the triggering of images for the integration, it only enables the integration process.
Background correction will wait after `BCIntegrationStart`, until `BCIntegrationFrameCount` frames have been produced by the camera.
Frame triggering is not in the background correction domain. This is controlled by features such as `ExposureTime`, `AcquisitionStart`, `AcquisitionStop`, `TriggerSource`, `TriggerSelector`, `AcquisitionFrameRate`, and etc. If the camera does not output images for some reason, background correction integration will stall until `AcquisitionStart` is executed and frame triggering is allowed by the trigger setup.

For optimal correction results:

1. Configure the settings you intend to use for your application.
2. Integrate a fresh background correction image without light (dark image) using these settings.
3. Finally, apply the background correction.

BCMode [Enum] R/W

Origin of feature: Camera

Controls the operating mode of the background correction. Different modes may be available, depending on the previously integrated corrected data.

`Off`	`[Default] Background correction is off`
`On`	Current correction image—if valid, see `BCState`—is subtracted from the live image and the `BCDatasetOffsetValue` is added
`OffsetOnly`	`BCDatasetOffsetValue` is added to the live image
`ReferenceImage`	Current correction image—if valid, see `BCState`—is output instead of the live image

BCState [Enum] R

Origin of feature: Camera

Shows the current state of the background correction processing.

`Ok`	`Background correction` is operating normally as configured with `BCMode`
`DatasetInvalid`	A new integration might be needed or is still in progress
`ROIOutOfBounds`	ROI settings might be out of the valid range of the integrated correction image

DefectPixelCorrection

DPCDatasetActivate [Command]

Origin of feature: Camera

Activates the dataset that is currently indexed by `DPCDatasetSelector`.

DPCDatasetActive [Integer] R

Origin of feature: Camera
Range: [0 – Camera dependent]

The index of the active dataset, starting at ***0***. The maximum possible value of `DPCDatasetActive` depends on the number of valid datasets in the camera. The mapping of an index value to a specific correction data file may vary from camera to camera or after correction data modifications.
Use the `DPCDatasetSelector` and corresponding features to retrieve more information about the datasets.

DPCDatasetActiveDescription [String] R

Origin of feature: Camera

Gives a short descriptive label to the dataset that is currently active and indexed by `DPCDatasetActive`.

DPCDatasetDescription [String] R

Origin of feature: Camera

Gives a short descriptive label to the dataset that is currently indexed by `DPCDatasetSelector`.

DPCDatasetSelector [Integer] R/W

Origin of feature: Camera	
Range: [0 – Camera dependent]	Default: ***Camera dependent***

Selects a dataset for access. The maximum possible value of `DPCDatasetSelector` depends on the number of valid datasets in the camera. The mapping of an index value to a specific correction data file may vary from camera to camera or after correction data modifications.

DPCMode [Enum] R/W

Origin of feature: Camera	
Possible values: ***Off***, ***On***	Default: ***On***

Configures operation mode of the defect pixel correction.

NonUniformityCorrection

NUCDatasetActivate [Command]

Origin of feature: Camera

Activates the dataset that is currently indexed by the `NUCDataSetSelector`.

NUCDatasetActive [Integer] R

Origin of feature: Camera

Range: [0 – Camera dependent]

The index of the active dataset, starting at ***0***. The maximum possible value depends on the number of valid datasets in the camera. The mapping of an index value to a specific correction data file may vary from camera to camera or after correction data modifications. Use the `NUCDatasetSelector` and corresponding features to retrieve more information about the datasets.

NUCDatasetActiveDescription [String] R

Origin of feature: Camera

Gives a short descriptive label to the dataset that is currently indexed by `NUCDatasetActive`. For example: Gain 0, 15.000°C, 1000µs.

This text is intended for informational purposes in the graphical user interface only.

For the actual values refer to `NUCDatasetActiveExposureTime`, `NUCDatasetActiveGain`, and `NUCDatasetActiveTemperature`

NUCDatasetActiveExposureTime [Float] R

Origin of feature: Camera

Shows exposure time at acquisition of the dataset that is currently indexed by `NUCDatasetActive`. The dataset should be selected so that the actual exposure time setting corresponds to the reference value.

The number of distinct reference values is limited by available correction data, depending on the camera model.

NUCDatasetActiveGain [Float] R

Origin of feature: Camera

`SensorGain` setting at acquisition of the dataset that is currently indexed by `NUCDatasetActive`. The dataset should be selected so that the actual sensor gain setting corresponds to the reference value.

0	`SensorGain = Gain0`
1	`SensorGain = Gain1`
2	`SensorGain = Gain2`

The number of distinct reference values is limited by available correction data, depending on the camera model.

NUCDatasetActiveTemperature [Float] R

Origin of feature: Camera

Shows sensor temperature, in °C, at acquisition of the dataset that is currently indexed by `NUCDatasetActive`. The dataset should be selected so that the actual sensor temperature is close to the reference temperature.

The number of distinct reference values is limited by available correction data, depending on the camera model.

NUCDatasetAuto [Enum] R/W

Origin of feature: Camera

Controls automatic selection of the `NUCDatasetActive`.

Off	[Default] The automatic mode is off
Once	Auto-NUC occurs until target is achieved, then `NUCDatasetAuto` returns to *Off*
Continuous	The non-uniformity correction will continue according to the scene illumination

NUCDatasetDescription [String] R

Origin of feature: Camera

Description of the dataset indexed by `NUCDatasetSelector`.

NUCDatasetExposureTime [Float] R

Origin of feature: Camera

Shows the exposure time at acquisition of the dataset indexed by `NUCDatasetSelector`. The dataset should be selected so that the actual exposure time setting corresponds to the reference value.

NUCDatasetGain [Float] R

Origin of feature: Camera

`SensorGain` setting at acquisition of the dataset indexed by `NUCDatasetSelector`. The dataset should be selected so that the actual sensor gain setting corresponds to the reference value.

0	`SensorGain = Gain0`
1	`SensorGain = Gain1`
2	`SensorGain = Gain2`

The number of distinct reference values is limited by available correction data, depending on the camera model.

NUCDatasetNodeSelector [Integer] R/W

Origin of feature: Camera

Range: [0 – Camera dependent]

Selects a data point of a dataset for access to its properties, starting at ***0***. The maximum possible value depends on the number of valid data points in the dataset.

NUCDatasetNodeValue [Float] R

Origin of feature: Camera

Shows the set value of the selected data point. Set point defines a mean value which the corrected image will have if the input image has a mean value of the corresponding correction data image.

NUCDatasetSelector [Integer] R/W

Origin of feature: Camera

Values: [0 – Camera dependent]	Default: ***`Camera dependent`***

Selects a dataset for access. The maximum possible value depends on the number of valid datasets in the camera. The mapping of an index value to a specific correction data file may vary from camera to camera or after correction data modifications.

NUCDatasetTemperature [Float] R

Origin of feature: Camera

Sensor temperature, in °C, at acquisition of the dataset indexed by `NUCDatasetSelector`. The dataset should be selected so that the actual sensor temperature is close to the reference temperature.

The number of distinct reference values is limited by available correction data, depending on the camera model.

NUCMode [Enum] R/W

Origin of feature: Camera

Controls the operating mode of the non-uniformity correction. Depending on the factory-provided correction data, different modes may be available.

`Off`	Non-uniformity correction is off
`OnePoint`	Only one reference point is used for correction
`TwoPoint`	[Default] Two reference points are used for correction
`ThreePoint`	Three reference points are used for correction

ImageFormatControl

BinningHorizontal [Integer] R/W

Origin of feature: Camera	
Range: [1 – Camera dependent]	Default: ***`1`***

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

Changing `BinningHorizontal` value may affect the effective ROI size and position.

BinningHorizontalMode [Enum] R/W

Origin of feature: Camera
Possible value: ***`Sum`***

Determines the mode of horizontal binning. Binning is accomplished by summing the charge / gray value of adjacent pixels on sensor.

BinningVertical [Integer] R/W

Origin of feature: Camera	
Range: [1 – Camera dependent]	Default: ***1***

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

Changing `BinningVertical` value may affect the effective ROI size and position.

BinningVerticalMode [Enum] R/W

Origin of feature: Camera
Possible value: ***Sum***

Determines the mode of vertical binning. Binning is accomplished by summing the charge / gray value of adjacent pixels on sensor.

Height [Integer] R/W

Origin of feature: Camera
Range: [Camera dependent]

Height of image.

HeightMax [Integer] R

Origin of feature: Camera

Maximum image height for the current image mode.

ImageSize [Integer] R

Origin of feature: Camera

Size of images, in bytes, for the current format and size.

OffsetX [Integer] R/W

Origin of feature: Camera	
Range: [0 – Camera dependent]	Default: ***0***

Starting column of the readout region (relative to the first column of the sensor) in pixels.

OffsetY [Integer] R/W

Origin of feature: Camera	
Range: [0 – Camera dependent]	Default: ***0***

Starting row of the readout region (relative to the first row of the sensor) in pixels.

PixelFormat [Enum] R/W

Origin of feature: Camera

There are various pixel data formats that Allied Vision GigE IR and scientific cameras can output. Not all cameras have every mode (see the Technical Manuals for details).

Mono8	Bit depth: 8. One pixel every byte. Monochrome. For color cameras with on-camera interpolation, luminance (Y) channel returned.
Mono12	Bit depth: 12. One pixel every two bytes, LSB aligned. Monochrome. For color cameras with on-camera interpolation, luminance (Y) channel is returned.
Mono12Packed	Bit depth: 12. 2 pixels of data every 3 bytes. Monochrome. Doesn't support odd Width x Height.
Mono14	Bit depth: 14. One pixel every two bytes, LSB aligned. Monochrome. For color cameras with on-camera interpolation, luminance (Y) channel is returned.

SensorBits [Integer] R/C

Origin of feature: Camera

Maximum bit depth of sensor.

SensorHeight [Integer] R/C

Origin of feature: Camera

The total number of pixel rows on the sensor.

SensorOffsetX [Integer] R/C

Origin of feature: Camera

Absolute starting column of the readout region relative to the first column of the sensor, in pixels.

SensorOffsetY [Integer] R/C

Origin of feature: Camera

Absolute starting row of the readout region relative to the first row of the sensor, in pixels.

SensorType [Enum] R/C

Origin of feature: Camera

Type of image sensor. Monochrome or Bayer-pattern color sensor type.

SensorWidth [Integer] R/C

Origin of feature: Camera

The total number of pixel columns on the sensor.

Width [Integer] R/W

Origin of feature: Camera	
Range: [Camera dependent]	Unit: Pixels

Width of image.

WidthMax [Integer] R

Origin of feature: Camera

Maximum image width for the current image mode.

Info

GevDeviceMACAddress [Integer] R/C

Origin of feature: Driver
Display name: Device MAC address

48-bit MAC address of the GVCP interface of the selected remote device.

DevicePartNumber [String] R/C

Origin of feature: Camera

Manufacturer’s part number.

FirmwareVerBuild [Integer] R/C

Origin of feature: Camera

Build information.

FirmwareVerMajor [Integer] R/C

Origin of feature: Camera

Major part of the firmware version number (part before the decimal).

FirmwareVerMinor [Integer] R/C

Origin of feature: Camera

Minor part of firmware version number (part after the decimal).

LUTControl

The use of a look-up table (LUT) allows any mapping function in the form Output = f(Input) to be applied on the pixel values at runtime. The table is stored in the camera. Depending on the camera implementation, more than one table may be available.

LUTBitDepthIn [Integer] R/C

Origin of feature: Camera

Bit depth of the input value of the LUT block.

LUTBitDepthOut [Integer] R/C

Origin of feature: Camera

Bit depth of the output value of the LUT block.

LUTDatasetActive [Integer] R

Origin of feature: Camera

`LUTDatasetSelector` value of the last loaded LUT dataset. If a user set is saved, this value defines which LUT is loaded at the next start-up of camera. Use the `LUTSelector` followed by `LUTDatasetLoad` to activate to different value. If at any time more than one LUT instance is available, use `LUTSelector` to get the active dataset information for the other LUTs.

LUTDatasetLoad [Command]

Origin of feature: Camera

Loads the LUT dataset from file into volatile LUT memory. The file to be loaded is defined by `LUTDatasetSelector`.

To load a dataset file into the volatile LUT memory:

1. Select the file to be loaded with `LUTDatasetSelector`.
2. Select the target LUT with `LUTSelector`.
3. Execute `LUTDatasetLoad` command.

The LUT is not loaded if an empty dataset file is selected.

To avoid temporary invalid image data, it is recommended to perform one of the following actions before executing `LUTDatasetLoad`:

- Stop image acquisition.
- Set `LUTEnable` to *`False`*.

LUTDatasetSave [Command]

Origin of feature: Camera

Stores the current LUT dataset from the volatile memory of the camera to a file. `LUTDatasetSelector` defines the file in which the LUT is saved.

To save a LUT to file:

1. Select the LUT to be saved with `LUTSelector`.
2. Select the target file with `LUTDatasetSelector`.
3. Execute `LUTDatasetSave` command.

Some datasets may be factory defined and cannot be overwritten.

LUTDatasetSelector [Integer] R/W

Origin of feature: Camera	
Range: [0 - Number of LUT file selectors]	Default: *`0`*

Connects the LUT datasets to corresponding `FileSelector` entries accessible via the `FileAccessControl` features. Typical `FileSelector` entry names are:

- LUT0, LUT1, and so on

or

- LUTUser0, LUTUser1, and so on

Mapping to the integer value `LUTDatasetSelector` is camera specific. Some files may be factory provided and access be limited to read-only.

`LUTDatasetSelector` may point to an empty or non-initialized `FileSelector` entry. If trying to load such a file, an error occurs.

LUTEnable [Boolean] R/W

Origin of feature: Camera	
Possible values: ***`True`***, ***`False`***	Default: ***`False`***

Activates or deactivates the LUT that is currently selected by `LUTSelector`.

To avoid temporary invalid image data, it is recommended to perform one of the following actions before modifying the contents of the volatile LUT memory:

- Stop image acquisition.
- Set `LUTEnable` to ***`False`***.

LUTIndex [Integer] R/W

Origin of feature: Camera	
Range: [0 – ($2^{\text{LUTBitDepthIn}}$ - 1)]	Default: ***`0`***

Controls the index (offset) to access a single table entry in the LUT via the `LUTValue` feature. The corresponding LUT is selected by `LUTSelector`.

LUTSelector [Enum] R/W

Origin of feature: Camera

Selects the LUT instance to control. The number of available LUT instances and how they are connected is camera specific.

`Luminance`	LUT is applied on the luminance channel

LUTValue [Integer] R/W

Origin of feature: Camera	
Range: [0 – ($2^{\text{LUTBitDepthOut}}$ - 1)]	Default: ***`0`***

Returns or sets the value of the table entry at `LUTIndex` of the LUT instance selected by `LUTSelector`.

To avoid temporary invalid image data, it is recommended to perform one of the following actions before modifying the contents of the volatile LUT memory:

- Stop image acquisition.
- Set `LUTEnable` to ***`False`***.

LUTValueAll [Register] R/W

Origin of feature: Camera

Allows access to the complete table that is currently selected by `LUTSelector`.

This register should be treated as a byte array. The size per entry is 2 bytes that stores a 16 bit unsigned integer. The maximum valid value is limited to $[(2^{LUTBitDepthOut}) - 1]$ and the byte order is low significant byte first (little-endian). The number of entries available in this field is $2^{LUTBitDepthIn}$.

Stream

Info

GVSPFilterVersion [String] R/C

Origin of feature: Driver
Display name: GVSP Filter Version

Version of the GVSP Filter driver.

Multicast

Multicast mode allows the camera to send image data to all hosts on the same subnet as the camera. The host computer (or Vimba Viewer application instance) 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.

Most GigE switches support a maximum PacketSize of 1500 in multicast mode.

When using clients with Linux, you have to configure the IP subsystem to process Multicast IP traffic.

MulticastEnable [Boolean] R/W

Origin of feature: Driver	
Display name: Multicast Enable	
Possible values: *`True`*, *`False`*	Default: *`False`*

Enables multicast mode. In multicast mode all computers on the same subnet as the camera can receive image data from the camera `MulticastIPAddress`.

MulticastIPAddress [Integer] R/W

Origin of feature: Driver	
Display name: Multicast IP Address	

Sets the multicast IP address.

Settings

Definition GVSP = GigE Vision Streaming Protocol

GVSPAdjustPacketSize [Command]

Origin of feature: Driver	
Display name: GVSP Adjust Packet Size	

Requests the packet size used to be adjusted automatically.

GVSPBurstSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Burst Size		
Range: [1–256]	Default: ***32***	Unit: GVSP Packets

Maximum number of GVSP packets to be processed in a burst.

GVSPDriverSelector [Enum] R/W

Origin of feature: Driver	
Display name: GVSP Driver Selector	
Possible values: ***Filter***, ***Socket***	Default: ***Filter***

Streaming driver to be used.

GVSPHostReceiveBuffers [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Host Receive Buffers	
Range: [256–2048]	Default: ***512***

Number of buffers to be used by the network socket. Only applicable when not using the filter driver.

GVSPMaxLookBack [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Max Look Back	
Range: [1–1024]	Default: ***30***

Size of the look back window, in packets, when determining if a stream packet is missing. When a stream packet arrives out of order, the driver skips back `GVSPMaxLookBack` 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.

GVSPMaxRequests [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Max Requests	
Range: [1–512]	Default: ***3***

The maximum number of resend requests that the host will attempt before marking a packet dropped.

GVSPMaxWaitSize [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Max Wait Size	
Range: [8–1024]	Default: ***100***

Maximum number of received GVSP packets following a resend request to wait before requesting again.

GVSPMissingSize [Integer] R/W

Origin of feature: Driver	
Display name: GVSP Missing Size	
Range: [0–1024]	Default: ***256***

Maximum number of simultaneous missing GVSP packets before dropping the frame (0 = OFF).

GVSPPacketSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Packet Size		
Range: [Camera dependent]	Default: ***8999***	Unit: Bytes

GVSP packet size.

GVSPTiltingSize [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Tilting Size		
Range: [0–1024]	Default: ***100***	

Maximum number GVSP packets received from a following frame before dropping the frame (0 = OFF).

GVSPTimeout [Integer] R/W

Origin of feature: Driver		
Display name: GVSP Timeout		
Range: [10–5000]	Default: ***70***	Unit: ms

End of stream timeout. If no stream packet received before **`GVSPTimeout`**, host requests resend, up to **`GVSPMaxRequests`** times. If still no packet received from camera, packet is marked as dropped.

Statistics

The packet counts in these statistics cover the image transport. Packets used for camera control or event data are not counted. All counters are reset at **`AcquisitionStart`**.

StatFrameRate [Float] R

Origin of feature: Driver
Display name: Stat Frame Rate

Rate at which the device is acquiring frames, derived from the frame timestamps.

StatFrameDelivered [Integer] R

Origin of feature: Driver
Display name: Stat Frames Delivered

Number of error-free frames captured since the start of imaging.

StatFrameDropped [Integer] R

Origin of feature: Driver
Display name: Stat Frames Dropped

Number of incomplete frames received by the host due to missing packets (not including shoved frames).

StatFrameRescued [Integer] R

Origin of feature: Driver
Display name: Stat Frames Rescued

Number of frames that initially had missing packets but were successfully completed after packet resend.

StatFrameShoved [Integer] R

Origin of feature: Driver
Display name: Stat Frames Shoved

Number of frames dropped because the transfer of a following frame was completed earlier.

StatFrameUnderrun [Integer] R

Origin of feature: Driver
Display name: Stat Frames Underrun

Number of frames missed due to the non-availability of a user supplied buffer.

StatLocalRate [Float] R

Origin of feature: Driver
Display name: Stat Local Rate

Inverse of time interval between the last two frames (faulty or not) received by the host. No averaging is performed.

In case of error-free frame reception, `StatLocalRate` is similar to `StatFrameRate`, except that the host clock is used instead of frame timestamps for measuring the time interval between frames. Otherwise, `StatLocalRate` and `StatFrameRate` may differ significantly.

StatPacketErrors [Integer] R

Origin of feature: Driver
Display name: Stat Packets Errors

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

StatPacketMissed [Integer] R

Origin of feature: Driver
Display name: Stat Packets Missed

Number of packets missed since the start of imaging.

If everything is configured correctly, this number should remain zero, or at least very low compared to `StatPacketReceived`.

StatPacketReceived [Integer] R

Origin of feature: Driver
Display name: Stat Packets Received

Number of error-free packets received by the driver since the start of imaging, this number should grow steadily during continuous acquisition.

StatPacketRequested [Integer] R

Origin of feature: Driver
Display name: Stat Packets Requested

Number of missing packets that were requested to be resent from the device.

If everything is configured correctly, this number should remain zero, or at least very low compared to `StatPacketReceived`.

StatPacketResent [Integer] R

Origin of feature: Driver
Display name: Stat Packets Resent

Number of packets resent by the camera since the start of imaging.

StatTimeElapsed [Float] R

Origin of feature: Driver
Display name: Stat Time Elapsed

Elapsed time (in seconds) since the streaming was started.

StreamInformation

StreamID [String] R/C

Origin of feature: Driver	
Display name: Stream ID	

Vimba V1.3 or later Device's unique ID for the stream.

StreamType [Enum] R/C

Origin of feature: Driver	
Display name: Stream Type	

Vimba V1.3 or later Identifies the transport layer technology of the stream.

TransportLayerControl

GigEVision

GevCurrentIPConfigurationDHCP [Boolean] R/W

Origin of feature: Camera	
Possible values: `True`, `False`	Default: `True`

Controls whether the DHCP IP configuration scheme is activated on the given logical link.

GevCurrentIPConfigurationLLA [Boolean] R/W

Origin of feature: Camera	
Possible values: `True`, `False`	Default: `True`

Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Currently as per the GigE Vision specification, LLA cannot be disabled.

GevCurrentIPConfigurationPersistentIP [Boolean] R/W

Origin of feature: Camera	
Possible values: `True`, `False`	Default: `False`

Controls whether the Persistent IP configuration scheme is activated on the given logical link.

GevInterfaceSelector [Integer] R/W

Origin of feature: Camera

Selects which logical link to control.

GevMACAddress [Integer] R

Origin of feature: Camera

MAC address of the link specified by `GevInterfaceSelector`.

PayloadSize [Integer] R

Origin of feature: Camera

Unit: Bytes

Total size of payload.

- If `ChunkModeActive` = *`True`*:
  `PayloadSize` = `ImageSize` + `NonImagePayloadSize` + `8`
- If `ChunkModeActive` = *`False`*:
  `PayloadSize` = `ImageSize`

StreamHold

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 [Integer] R

Origin of feature: Camera

Unit: Frames

The maximum number of images (for the current size and format) that can be stored on the camera when `StreamHold` is enabled. Used when `AcquisitionMode` = *`Recorder`*, or `StreamHoldEnable` = *`On`*. This value is different for each camera depending on the camera internal memory size and the `ImageSize`.

StreamHoldEnable [Enum] R/W

Origin of feature: Camera

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

UserSetControl

UserSetDefaultSelector [Enum] R/W

Origin of feature: Camera

Possible values: ***Default***, ***UserSet1***, ***UserSet2***, ***UserSet3***, ***UserSet4***

Selects the user set to be loaded on power-up or reset.

UserSetLoad [Command]

Origin of feature: Camera

Loads camera parameters from the user set specified by `UserSetSelector`.

UserSetSave [Command]

Origin of feature: Camera

Saves camera parameters to the user set specified by `UserSetSelector`. The ***Default*** setting cannot be overwritten.

UserSetSelector [Enum] R/W

Origin of feature: Camera

Possible values: ***Default***, ***UserSet1***, ***UserSet2***, ***UserSet3***, ***UserSet4***

Selects a user set, for loading or saving camera parameters.

Index (GigE camera and driver features)

F

G

T

U

V

W

Z

Index (GigE IR & scientific camera and driver features)

F

G

H

I

J

L

M

N

O

P

R

S

T

U

W