GenICam

Standard Features Naming Convention

Version 1.2.1

GEN<i>CAM

2008-08-19

Table of Contents

History

Version	Date	Changed by	Change
0.01	14.02.2006	Eric Carey, DALSA Coreco	Initial version based on the GenICam standard feature list document of the GigE Vision/GenICam joint sub-committee. This version is intended to be the official feature naming convention to be used for GigE Vision cameras. Original contributors: **Basler** (Fritz Dierks, Thies Moeller, Andreas Gäer), **Leutron Vision** (Jan Becvar), **DALSA Coreco** (Eric Carey), **Euresys** (Jean-Michel Wintgens), **MVTec** (Christoph Zierl), **National Instruments** (Chris Graf), **Stemmer** (Sascha Dorenbeck), **SICKIVP** (Mattias Johannesson), **JAI** (Ole Krogh Jørgensen), **Matrox** (Stephane Maurice)
0.02	16.03.2006	Stephane Maurice, Matrox	Define the new Acquisition, Trigger and I/O feature set. Introduced the notion of counters and grouped it with Timers in a separate section. Reviewed feature names for consistency and grouping.
draft 1.00	04.04.2006	Stephane Maurice, Matrox	Included modifications and corrections based on the feedbacks from version 0.02 to 0.9. Final Draft.

Version	Date	Changed by	Change
draft 1.00.01	06.06.2006	Stephane Maurice, Matrox	Changed PixelSize to Bpp8, Bpp10, ... Removed all "_" in enumerations and all feature names.
draft 1.00.02	22.06.2006	Stephane Maurice, Matrox	Changed Software Trigger from TriggerMode to TriggerSource to permit 1394 DCAM feature compatibility. Removed Ticks as standard unit for Raw time unit. Added AnyEdge as standard signal activation and event type. Added Line0 and UserBit0 as standard names for enumeration. Added AcquisitionFrameRateRaw and AcquisitionLineRateRaw. Defined standard Event numbers that matches the GigEvision Event numbers.
draft 1.00.03	16.06.2007	Vincent Rowley, Pleora Technologies Inc.	Prepared Version 1.0. Removed the AIA logo. Fixed typos. Added a note with respect to how the GevMACAddress feature should be implemented. Added a note specifying that the GevCurrentIPConfiguration feature should not be used in production GenICam XML files since it will be deprecated in the next version of the present document. Fixed GevTimestampTickFrequency valid range.

Version	Date	Changed by	Change
draft 1.00.03 cont.	19.06.2007	Stephane Maurice, Matrox	Preparation for Version 1.0 continued: Added a note about the Selector usage specifying that they must not introduce side effect when their value is changed. Removed GiGEVision logo since the Standard Feature List is now part of the GenICam standard. Specified that features with big value such as GevMACAdress, GEVTimestampTickFrequency and GEVTimestampValue must be returned as a single 64 bit values.
Release 1.00.00	20.06.2007	Stephane Maurice, Matrox	Final release Version 1.00 Note: This release includes all the features as they were defined in the draft 1.00.02 referenced in the final GigE Vision specification version 1.00.
Version 1.01.01	04.07.2007	Vincent Rowley, Pleora Technologies Inc.	Added SensorTaps, SensorDigitizationTaps, GevCurrentIPConfigurationLLA, GevCurrentIPConfigurationDHCP, GevCurrentIPConfigurationPersistentIP and GevIPConfigurationStatus features. Deprecated GevCurrentIPConfiguration. Added OpenAccess to the list of valid values for the GevCCP feature.
Version 1.01.02	24.07.2007	Stephane Maurice Matrox	Added the PixelFormat description chapter and note about zero based user bits.
Release 1.1	2.10.2007	Stephane Maurice, Matrox	Final release Version 1.1

Version	Date	Changed by	Change
Version 1.1.01	10.09.2007	Thies Möller, Basler	Created chapter for File Access.
Version 1.1.02	12.01.2008	Stephane Maurice, Matrox Vincent Rowley , Pleora	Review and modification to the File Access features proposal.
Release 1.2	29.04.2008	Stephane Maurice, Matrox	SFNC 1.2 including the File Access features and corrections. Also removed the PixelFormat description chapter and GEV event numbers.
Version 1.2.01	17.07.2008	Karsten Ingeman Christensen, JAI	Merged with recommended visibility proposal from JAI and commented by Vincent Rowley, Pleora
Release 1.2.1	19.08.2008	Stephane Maurice, Matrox	SFNC 1.2.1 including the recommended visibility.

1 Introduction

The GenICam technology allows exposing arbitrary features of a camera through a unified API and GUI. Each feature can be defined in an abstract manner by its name, interface type, unit of measurement and behavior. The GenApi module of the GenICam standard defines how to write a camera description file that describes a specific camera's mapping.

GenICam alone is sufficient if the user wants to write software for a specific camera only because all features of the camera are accessible through the GenICam provided API. However if the user wants to write **generic software** for a whole class of cameras then GenICam alone is not sufficient. In addition, the software vendors and the camera vendors have to agree on a common naming convention for the standard features.

For technical and historical reasons the different transport layer technologies (GigE, 1394, Camera Link, etc.) might require slightly different feature sets. This is why this document addresses mainly the cameras compliant to the GigE Vision standard. The naming convention is however targeting maximum reusability by other existing and future transport layer technologies. It provides the definitions of **standard use cases** and **standard features**. The goal is to cover and to standardize the naming convention used in all those basic use cases where the implementation by different vendors would be very similar anyway.

Features are tagged within this document according to the following list:

- M: **mandatory -** Must be implemented to achieve compliance with the GigE Vision standard

- R: recommended **-** This feature adds important aspects to the use case and should respect the naming convention.*

- O: optional - This feature is less critical. Nevertheless, it is considered and should respect the naming convention.

For additional details about the mandatory features please refer to the GigE Vision standard.

Recommended Visibility

According to the GenICam standard each feature can be assigned a "recommended visibility" using the <Visibility> element in the XML-files. The <Visibility> element defines the user level that should get access to the feature. Possible values are: Beginner, Expert, Guru and Invisible. The latter is required to make features show up in API, but not in the GUI.

The visibility does not affect the functionality of the features but is merely used by the GUI to decide which features to display based on the current user level. The purpose is mainly to insure that the GUI is not cluttered with information that is not intended at the current user level.

The following criteria's have been used for the assignment of recommended visibility:

- B: beginner - features that should be visible for ***all*** users via the GUI and API. This is the default visibility in the GenICam XML-files and will be used if the <Visibility> element is omitted. The number of features with "beginner" visibility should be limited to all **basic** features of the devices so the GUI display is well-arranged and is easy to use.

- E: expert - features that require a more in-depth knowledge of the camera functionality. This is the preferred visibility level for all advanced features in the cameras.

- G: guru – advanced features that might bring the cameras into a state where it will not work properly anymore if it is set incorrectly for the cameras current mode of operation.

- I: invisible – features that should be kept hidden for the GUI users but still be available via the API.

1.1 Conventions

Selector

A selector is used to index which instance of the feature is accessed in situations where multiple instances of a feature exist (for instance, the analog gain for each separate channel for the red/green/blue component of a color camera). The selector is a separate feature that is typically an IEnumeration or an IInteger.

Features dependent on the Selector are expressed using the C language convention for arrays: a pair of brackets follows the feature name, like in SelectedFeature[Selector]. When the Selector is not present, one must deduce the feature is not an array.

Note that selectors must be used only to select the target features for subsequent changes. It is not allowed to change the behavior of a device in response to a change of a selector value.

Standard Units

The following abbreviations are used as standard units for features described in this document. Note that all units are using plan ASCII characters.

us	microseconds
ms	milliseconds
s	seconds
dB	decibels
C	Celsius

Hz	Hertz

1.2 Acronyms

ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AIA	Automated Imaging Association
AOI	Area Of Interest
CRT	Cathode Ray Tube
DC	Direct Current
DHCP	Dynamic Host Configuration Protocol
EMVA	European Machine Vision Association
ID	Identifier
I/O	Input/Output
IP	Internet Protocol
LLA	Link-Local Address
LUT	Look-Up Table
M	Mandatory
O	Optional
R	Recommended or Read (depends on the context)
ROI	Region Of Interest
URL	Uniform Resource Locator
W	Write
XML	eXtensible Markup Language

2 Feature Summary

This section provides a comprehensive list of all standard features covered by this document. The following sections provide in-depth explanation of each feature. Note that, in case of discrepancy, those chapters describing the features in detail prevail.

For this section, mandatory features are listed in **bold**. Optional features are listed in *italic*.

2.1 Device Information

Device Information provides description of the camera and its sensor.

Table 2-1: Device Information Summary

Name	Level	Interface	Access	Unit	Visibility	Description
DeviceVendorName	R	IString	R	-	B	The name of the device vendor.
DeviceModelName	R	IString	R	-	B	The name of the device model.
DeviceManufacturerInfo	R	IString	R	-	B	Additional info from manufacturer about this device.
DeviceVersion	R	IString	R	-	B	A string identifying the version of the device.
DeviceFirmwareVersion	R	IString	R	-	B	Version of firmware/software.
DeviceID	R	IString	R	-	E	A unique identifier of the device, e.g., a serial number or a GUID (User Data in GigE Boot register).
DeviceUserID	O	IString	R/W	-	B	A user set ID that is user-programmable.
DeviceScanType	R	IEnumeration	R/(W)	-	E	{Areascan, Linescan} Typically only Read, can be Write for setup-image/measurement selection.
DeviceMaxThroughput	O	IInteger	R	bytes/sec	E	Maximum number of bytes per second device supports.
DeviceReset	R	ICommand	W	-	G	Resets and reboots the device immediately.
DeviceRegistersStreamingStart	R	ICommand	W	-	G	Announces the start of registers streaming without immediate checking

						for consistency.
DeviceRegistersStreamingEnd	R	ICommand	W	-	G	Announces the end of registers streaming and perform validation for registers consistency before activating them. This will also update the DeviceRegistersValid flag.
DeviceRegistersCheck	R	ICommand	W	-	E	Performs an explicit register set validation for consistency.
DeviceRegistersValid	R	IBoolean	R	-	E	Informs whether the current register set is valid and consistent.

2.2 Image Format Control

Image Format Control lists all features controlling the size of the transmitted image.

Table 2-2: Image Format Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
SensorWidth	R	IInteger	R	pixels	E	Width of sensor (effective pixels)
SensorHeight	R	IInteger	R	pixels	E	Height of sensor (effective pixels)
SensorTaps	O	IEnumeration	R/(W)	-	E	Number of taps of the camera sensor.
SensorDigitizationTaps	O	IEnumeration	R/(W)	-	E	Number of digitized samples outputted simultaneously by the camera A/D conversion stage.
WidthMax	R	IInteger	R	pixels	E	Maximum image width. Depends on binning & decimation.
HeightMax	R	IInteger	R	pixels	E	Maximum image height. Depends on binning & decimation.
Width	M	IInteger	R/(W)	pixels	B	Width of Image/Area Of Interest.
Height	M	IInteger	R/(W)	pixels	B	Height of Image/Area Of Interest.
OffsetX	R	IInteger	R/W	pixels	B	X offset or left coordinate of the Area Of Interest.
OffsetY	R	IInteger	R/W	pixels	B	Y offset or top coordinate of the Area Of Interest.

LinePitch	R	IInteger	R/W	bytes	E	Distance between consecutive lines in bytes.
BinningHorizontal	O	IInteger	R/W	cells	E	Number of horizontally binned cells (1 = no binning).
BinningVertical	O	IInteger	R/W	cells	E	Number of vertically binned cells (1 = no binning).
DecimationHorizontal	O	IInteger	R/W	cells	E	Number of horizontally skipped cells (0 = no skipping)
DecimationVertical	O	IInteger	R/W	cells	E	Number of vertically skipped cells (0 = no skipping)
ReverseX	R	IBoolean	R/W	-	E	When set to true, this parameter flips the image horizontally.
ReverseY	R	IBoolean	R/W	-	E	When set to true, this parameter flips the image vertically.
PixelCoding	R	IEnumeration	R/(W)	-	E	Color-coding of the image pixels. {Mono, MonoSigned, MonoPacked, RGBPacked, BGRPacked, RGBAPacked, BGRAPacked, RGBPlanar, YUV411Packed, YUV422Packed, YUV444Packed, Raw}
PixelSize	R	IEnumeration	R/(W)	-	E	Size in bits of the image pixels. {Bpp8, Bpp10, Bpp12, Bpp14, Bpp16, Bpp24, Bpp32, Bpp48, Bpp64}
PixelColorFilter	R	IEnumeration	R	-	E	Color filter convention of the image. {None, BayerRG, Bayer GB, BayerGR, BayerBG}
PixelFormat	M	IEnumeration	R/(W)	-	B	Format of the image pixels. For more details, see the Pixel Format description chapter. {Mono8, Mono8Signed, Mono10, Mono10Packed, Mono12, Mono12Packed, Mono16, BayerGR8, BayerRG8, BayerGB8, BayerBG8, BayerGR10, BayerRG10, BayerGB10, BayerBG10, BayerGR12, BayerRG12, BayerGB12, BayerBG12, RGB8Packed, BGR8Packed, RGBA8Packed, BGRA8Packed, RGB10Packed, BGR10Packed, RGB12Packed, BGR12Packed, RGB10V1Packed, RGB10V2Packed, YUV411Packed, YUV422Packed, YUV444Packed, RGB8Planar, RGB10Planar, RGB12Planar, RGB16Planar}
PixelDynamicRangeMin	O	IInteger	R/W	-	E	Minimum pixel value sent by the camera.
PixelDynamicRangMax	O	IInteger	R/W	-	E	Maximum pixel value sent by the camera.

TestImageSelector	O	IEnumeration	R/W	-	B	Selection of the test image to be used. {Off, Black, White, …}

2.3 Acquisition and Trigger Controls

Acquisition and Trigger Controls lists all features that relate to actual image acquisition, including the triggering mode.

Table 2-3: Acquisition and Trigger Controls Summary

Name	Level	Interface	Access	Unit	Visibility	Description
AcquisitionMode	M	IEnumeration	R/W	-	B	Defines the type of acquisition. {SingleFrame, MultiFrame, Continuous}
AcquisitionStart	M	ICommand	W	-	B	Issues the START command. This starts the acquisition.
AcquisitionStop	M	ICommand	W	-	B	Issues the STOP command. This stops the acquisition.
AcquisitionAbort	R	ICommand	W	-	E	Issues the ABORT command. This immediately aborts the acquisition without completing the current frame.
AcquisitionArm	O	ICommand	W	-	E	Issues the acquisition Arm command. This optional command validates all the current features consitency and prepares the device for a fast AcquisitionStart.
AcquisitionFrameCount	R	IInteger	R/W	frames	B	Number of frames to be acquired in MultiFrame acquisition mode. The minimum allowable value is 1.
AcquisitionFrameRateAbs	R	IFloat	R/W	Hz	B	Controls the desired frame rate of the imager. Applicable when the imager can be configured to output images at a given rate. Absolute units are in Hz.
AcquisitionFrameRateRaw	O	IInteger	R/W	-	B	Controls the desired frame rate of the imager. Applicable when the imager can be configured to output images at a given rate. Units are device specific.
AcquisitionLineRateAbs	O	IFloat	R/W	Hz	B	Controls the desired line rate of the imager. Applicable when the imager can be configured to output lines at a given rate.

						Units are device specific. Absolute units are in Hz.
AcquisitionLineRateRaw	O	IInteger	R/W	-	B	Controls the desired line rate of the imager. Applicable when the imager can be configured to output lines at a given rate. Units are device specific.
AcquisitionStatusSelector	R	IEnumeration	R/W	-	E	Selector for AcquisitionStatus to read. {FrameTriggerWait, FrameActive, FrameTransfer, ExposureActive, AcquisitionTriggerWait, AcquisitionActive, AcquisitionTransfer, ….}
AcquisitionStatus[AcquisitionStatusSelector]	R	IBoolean	R	-	E	Status of the selected acquisition flag.
TriggerSelector	R	IEnumeration	R/W	-	B	Select the type of trigger to control. {FrameStart, FrameEnd, FrameActive, AcquisitionStart, AcquisitionEnd, AcquisitionActive, LineStart, ExposureStart, ExposureEnd, ExposureActive,…}.
TriggerMode[TriggerSelector]	R	IEnumeration	R/W	-	B	Specifies the operation mode of the trigger for the acquisition. {Off, On}.
TriggerSoftware[TriggerSelector]	R	ICommand	W	-	B	Generates a software trigger to start the acquisition when trigger mode is active and trigger source is Software.
TriggerSource[TriggerSelector]	R	IEnumeration	R/W	-	B	Specifies the source for the trigger (input line signal, software, …). {Software, Line0 (if 0 based), Line1, …, Timer1Start, Counter1End, …}
TriggerActivation[TriggerSelector]	R	IEnumeration	R/W	-	B	Specifies the type of signal change that will activate the trigger {RisingEdge, FallingEdge, AnyEdge, LevelHigh, LevelLow, …}.
TriggerOverlap[TriggerSelector]	O	IEnumeration	R/W	-	E	Selects which type of trigger will be considered valid when it overlaps with a part of a previous frame such as CCD readout or exposure.

						{Off, ReadOut, PreviousFrame}
TriggerDelayAbs[TriggerSelector]	O	IFloat	R/W	us	E	Selects the absolute delay in microseconds to apply after reception of the trigger signal before starting exposure for the acquisition.
TriggerDelayRaw[TriggerSelector]	O	IInteger	R/W	-	E	Selects the raw delay in device specific unit to apply after reception of the trigger signal before starting exposure for the acquisition.
TriggerDivider[TriggerSelector]	O	IInteger	R/W	-	E	Scale factor for the acquisition trigger. Allows scaling down the trigger frequency.
TriggerMultiplier[TriggerSelector]	O	IInteger	R/W	-	E	Scale factor for the acquisition trigger. Allows scaling up the trigger frequency.
ExposureMode	R	IEnumeration	R/W	-	B	Mode of operation for the exposure control (or shutter). {Off, Timed, TriggerWidth, TriggerControlled, …}
ExposureTimeAbs	R	IFloat	R/W	us	B	Controls the Absolute exposure time in microseconds (us).
ExposureTimeRaw	O	IInteger	R/W	-	B	Controls the Raw exposure time in device-specific units.
ExposureAuto	O	IEnumeration	R/W	-	B	Auto-adjustment of the exposure time. {Off, Once, Continuous}

2.4 Digital I/O

Digital I/O describes the features required to control the general input and output pins of the camera.

Table 2-4: Digital I/O Summary

Name	**Level**	**Interface**	**Access**	**Unit**	**Visibility**	**Description**
LineSelector	R	IEnumeration	R/W	-	E	Selector for the physical line (or pin) for the Input/Output signal. { Line0 (if 0 based), Line1, Line2, …}

LineMode [LineSelector]	R	IEnumeration	R/W	-	E	Select the Line mode. {Input, Output}
LineInverter[LineSelector]	R	IBoolean	R/W	-	E	State of the Line input or output Inverter associated with the physical line.
LineStatus[LineSelector]	R	IBoolean	R	-	E	Current logical state of signal at time of polling.
LineStatusAll	O	IInteger	R	bitfield	E	Current logical state of all available Line signals at time of polling in a single bitfield. The order is Line0 (if 0 based), Line1, Line2, …
LineSource[LineSelector]	R	IEnumeration	R/W	-	E	Source signal for the Output Line. {Off, AcquisitionActive, FrameActive, ExposureActive, TimerActive(n) , CounterActive(n), UserOutput(n), …}
LineFormat[LineSelector]	O	IEnumeration	R/W	-	E	Configuration of the physical line. {NoConnect, TriState, OptoCoupled, TTL, LVDS, RS422, ...}
UserOutputSelector	R	IEnumeration	R/W	-	E	Selector for User output bits. { UserOutput0 (if 0 based),, UserOutput1, …}
UserOutputValue[UserOutputSelector]	R	IBoolean	R/W	-	E	Set a single user output bit value.
UserOutputValueAll	O	IInteger	R/W	bitfield	E	Set all user output bits in one access.
UserOutputValueAllMask	O	IInteger	R/W	bitfield	E	Defines mask to use when setting all the User output bits in one access.

2.5 Counters and Timers

The Counters and Timers section describes the features required to control the usage of programmable counters and timers.

Table 2-5: Counters and Timers Summary

Name	Level	Interface	Access	Unit	Visibility	Description

CounterSelector	R	IEnumeration	R/W	-	E	Selects the Counter to configure. {Counter1, Counter2, …}
CounterEventSource[CounterSelector]	R	IEnumeration	R/W	-	E	Source for the Events to count. {Off, AcquisitionTrigger, FrameStart, FrameEnd, or any acquisition Events, Line0RisingEdge (if 0 based), Line1RisingEdge, Line2RisingEdge, …, Counter1End, …., Timer1End, … TimestampTick, ... }
CounterReset[CounterSelector]	R	ICommand	W	-	E	Resets the selected Counter.
CounterValue[CounterSelector]	R	IInteger	R	-	E	Reads the current count of the selected Counter.
CounterValueAtReset[CounterSelector]	R	IInteger	R	-	E	Reads the count of the selected Counter when it was reset.
CounterDuration[CounterSelector]	R	IInteger	R/W	-	E	Specifies the Duration (or end value) for the selected counter. A CounterEnd Event is generated and the CounterActive flag is reset when the target count is reached.
CounterStatus[CounterSelector]	R	IEnumeration	R	-	E	Reports the actual status of the selected Counter. {CounterIdle, CounterTriggerWait, CounterActive, CounterCompleted, CounterOverflow}
CounterTriggerSource[CounterSelector]	R	IEnumeration	R/W	-	E	Source of the Trigger that starts or enables the Counter increment. {Off, AcquisitionStart, FrameStart, FrameEnd, ExposureStart, or any acquisition event, Line0 (if 0 based), Line1,… , Counter2End, …, ExposureActive or other internal Status signal used to control the counter duration }
CounterTriggerActivation[CounterSelector]	R	IEnumeration	R/W	-	E	Specifies the type of signal that will trigger or enable Counter increment. {RisingEdge, FallingEdge, AnyEdge, LevelHigh, LevelLow}
TimerSelector	R	IEnumeration	R/W	-	E	Selects which Timer to configure. {Timer1, Timer2, …}
TimerDurationAbs[TimerSelector]	R	IFloat	R/W	us	E	Length of the output (strobe) pulse in microseconds.

TimerDurationRaw[TimerSelector]	R	IInteger	R/W	-	E	Length of the output (strobe) pulse in device specific units.
TimerDelayAbs[TimerSelector]	R	IFloat	R/W	us	E	The absolute delay in microseconds from the TimerTrigger to the actual Timer pulse output.
TimerDelayRaw[TimerSelector]	R	IInteger	R/W	-	E	The raw Delay in device-specific unit from the TimerTrigger to the actual Timer pulse output.
TimerValueAbs[TimerSelector]	R	IFloat	R	us	E	Reports the actual value of the selected Timer in microseconds.
TimerValueRaw[TimerSelector]	R	IInteger	R	-	E	Reports the actual value of the selected Timer in device-specific unit.
TimerStatus[TimerSelector]	R	IEnumeration	R	-	E	Reports the actual status of the selected Timer. {TimerIdle, TimerTriggerWait, TimerActive, TimerCompleted}
TimerTriggerSource[TimerSelector]	R	IEnumeration	R/W	-	E	Selects which internal signal will trigger the Timer. {Off, AcquisitionTrigger, ExposureStart, Line0 (if 0 based), Line1, …, UserOutput0, UserOutput1, …}
TimerTriggerActivation[TimerSelector]	R	IEnumeration	R/W	-	E	Specifies the type of signal that will trig or enable Timer. {RisingEdge, FallingEdge, AnyEdge, LevelHigh, LevelLow}

2.6 Event Generation

The Event Generation section describes the features required to control the generation by the device of Event notifications sent to the host application.

Table 2-6: Event Generation Summary

Name	Level	Interface	Access	Unit	Visibility	Description
EventSelector	R	IEnumeration	R/W	-	E	Selector for the Event to control. An Event reflects the change of an internal signal state.. {AcquisitionTrigger, AcquisitionStart, AcquisitionEnd, FrameTrigger, FrameStart, FrameEnd, FrameTransferStart, FrameTransferEnd,

						ExposureStart, Exposure End, Counter1Start, Counter1End,…, Timer1Start, Timer1End,…, Line0RisingEdge, Line1RisingEdge, …, Line0FallingEdge, Line1FallingEdge, …, Line0AnyEdge, Line1AnyEdge, …}
EventNotification[EventSelector]	R	IEnumeration	R/W	-	E	Notification type issued when event occurs. {Off, GigEVisionEvent, …)

2.7 Analog Controls

Analog Controls lists all features related to the video signal in the analog domain (before digitization).

Table 2-7: Analog Controls Summary

Name	Level	Interface	Access	Unit	Visibility	Description
GainSelector	O	IInteger	R/W	-	B	Selects which Gain to control. {All, Red, Green, Blue, Tap1, Tap2, …}
GainRaw[GainSelector]	O	IInteger	R/W	-	B	Controls the selected Gain as integer.
GainAbs[GainSelector]	O	IFloat	R/W	-	B	Controls the seletcted Gain as float.
GainAuto[GainSelector]	O	IEnumeration	R/W	-	B	Performs auto adjustment of the selected Gain (AGC). {Off, Once, Continuous}
GainAutoBalance	O	IEnumeration	R/W	-	B	Performs automatic Gain balancing between sensor taps. {Off, Once, Continuous}
BlackLevelSelector	O	IInteger	R/W	-	E	Selects which tap is controlled by BlackLevelRaw and BlackLevelAbs. {All, Red, Green, Blue, Tap1, Tap2, …}
BlackLevelRaw[BlackLevelSelector]	O	IInteger	R/W	-	E	Controls the black level (offset) as integer.
BlackLevelAbs[BlackLevelSelector]	O	IFloat	R/W	-	E	Controls the black level (offset) as float.

BlackLevelAuto[BlackLevelSelector]	O	IEnumeration	R/W	-	E	Performs auto calibration of black level (offset). {Off, Once, Continuous}
BlackLevelAutoBalance	O	IEnumeration	R/W	-	E	Performs automatic BlackLevel (offset) balancing between taps. {Off, Once, Continuous}
WhiteClipSelector	O	IInteger	R/W	-	E	Selects which tap WhiteClip controls. {All, Red, Green, Blue, Tap1, Tap2, …}
WhiteClipRaw[WhiteClipSelector]	O	IInteger	R/W	-	E	Sets the selected white clip value as integer.
WhiteClipAbs[WhiteClipSelector]	O	IFloat	R/W	-	E	Sets the selected white clip value as float.
BalanceRatioSelector	O	IEnumeration	R/W	-	E	Selects the balance ratio to control. {Red, Green, Blue, Y, U, V, …}
BalanceRatioAbs[BalanceRatioSelector]	O	IFloat	R/W	-	E	Represents the ratio of the selected color component to a reference color component.
BalanceWhiteAuto	O	IEnumeration	R/W	-	E	Perfoms automatic white balancing. {Off, Once, Continuous}
Gamma	O	IFloat	R/W	-	B	Used for arbitrary gamma correction.

2.8 LUT Controls

LUT Controls provides all features related to look-up table.

Table 2-8: Lut Controls Summary

Name	Level	Interface	Access	Unit	Visibility	Description
LUTSelector	O	IEnumeration	R/W	-	E	Selection of the LUT to control. {Luminance, Red, Green, Blue, …}
LUTEnable[LUTSelector]	O	IBoolean	R/W	-	E	Enables the selected LUT.

LUTIndex[LUTSelector]	O	IInteger	R/W	-	G	Index of LUT element to access. This value is used by the SwissKnife to index into an array.
LUTValue[LUTSelector][LUTIndex]	O	IInteger	R/W	-	G	Value of selected LUT element found at index LutIndex.
LUTValueAll[LUTSelector]	O	IRegister	R/W	-	G	Accesses the whole content of selected LUT in one chunk access.

2.9 GigE Vision Transport Layer

GigE Vision Transport Layer lists all the features related to the GigE Vision transport specification.

Table 2-9: GigE Vision Transport Layer Summary

Name	Level	Interface	Access	Unit	Visibility	Description
PayloadSize	M	IInteger	R	bytes	E	Size of the payload in bytes. This is the total number of bytes sent in the payload. Image data + chunk data if present. No packet headers.
GevVersionMajor	R	IInteger	R	-	E	This field represents the major version of the GigE Vision specification supported by this device. 1.0 for the first release of GigE Vision.
GevVersionMinor	R	IInteger	R	-	E	This field represents the minor version of the GigE Vision specification supported by this device. 1.0 for the first release of GigE Vision.
GevDeviceModeIsBigEndian	O	IBoolean	R	-	G	Endianess might be used to interpret multi-byte data for READMEM and WRITEMEM commands. This represents the endianess of bootstrap registers. FALSE: Little-endian device TRUE: Big-endian device

GevDeviceModeCharacterSet	O	IEnumeration	R	-	G	This feature represents the character set. It must take one of the following values. Note: only UTF8 is supported by GigE Vision 1.0. {UTF8}
GevInterfaceSelector	O	IInteger	R/W	-	E	Indicates the index of the network interface to configure
GevMACAddress[GevInterfaceSelector]	O	IInteger	R	-	B	48-bit MAC address of the selected interface
GevSupportedIPConfigurationLLA[GevInterfaceSelector]	O	IBoolean	R	-	E	Indicate if LLA (Auto-IP) is supported by the selected interface
GevSupportedIPConfigurationDHCP[GevInterfaceSelector]	O	IBoolean	R	-	E	Indicate if DHCP is supported by the selected interface
GevSupportedIPConfigurationPersistentIP[GevInterfaceSelector]	O	IBoolean	R	-	E	Indicate if Persistent IP is supported by the selected interface
GevCurrentIPConfiguration[GevInterfaceSelector]	O	IEnumeration	R/W	-	B	Currently used IP configuration scheme for the selected interface. {PersistentIP, DHCP, LLA}
GevCurrentIPConfigurationLLA	O	IEnumeration	R/W	-	B	Indicates if the LLA IP configuration scheme is activated on the selected network interface.
GevCurrentIPConfigurationDHCP	O	IEnumeration	R/W	-	B	Indicates if the DHCP IP configuration scheme is activated on the selected network interface.
GevCurrentIPConfigurationPersistentIP	O	IEnumeration	R/W	-	B	Indicates if the Persistent IP configuration scheme is activated on the selected network interface.
GevCurrentIPAddress[GevInterfaceSelector]	O	IInteger	R	-	B	IP address of the selected interface.
GevCurrentSubnetMask[GevInterfaceSelector]	O	IInteger	R	-	B	Subnet mask of the selected interface.

GevCurrentDefaultGateway[GevInterfaceSelector]	O	IInteger	R	-	B	Default gateway of the selected interface.
GevFirstURL	O	IString	R	-	G	NULL-terminated string providing the first URL to the XML device description file.
GevSecondURL	O	IString	R	-	G	NULL-terminated string providing the second URL to the XML device description file.
GevNumberOfInterfaces	O	IInteger	R	-	E	Indicates the number of physical network interfaces on this device. A device must have at least one network interface.
GevPersistentIPAddress[GevInterfaceSelector]	O	IInteger	R/W	-	E	Persistent IP address for the selected interface. Only available if Persistent IP is supported by the device.
GevPersistentSubnetMask [GevInterfaceSelector]	O	IInteger	R/W	-	E	Persistent subnet mask for the selected interface. Only available if Persistent IP is supported by the device.
GevPersistentDefaultGateway[GevInterfaceSelector]	O	IInteger	R/W	-	E	Persistent default gateway for the selected interface. Only available if Persistent IP is supported by the device.
GevMessageChannelCount	O	IInteger	R	-	E	Indicates the number of message channels supported by this device. It can take two values: 0 or 1.
GevStreamChannelCount	O	IInteger	R	-	E	Indicates the number of stream channels supported by this device. It can take any value from 1 to 512.
GevSupportedOptionalCommandsUserDefinedName	O	IBoolean	R	-	G	Indicates if the User-defined Name register is supported.

GevSupportedOptionalCommandsSerialNumber	O	IBoolean	R	-	G	Indicates if the Serial Number register is supported.
GevSupportedOptionalCommandsEVENTDATA	O	IBoolean	R	-	G	Indicates if EVENTDATA_CMD and EVENTDATA_ACK are supported.
GevSupportedOptionalCommandsEVENT	O	IBoolean	R	-	G	Indicates if EVENT_CMD and EVENT_ACK are supported.
GevSupportedOptionalCommandsPACKETRESEND	O	IBoolean	R	-	G	Indicates if PACKETRESEND_CMD is supported.
GevSupportedOptionalCommandsWRITEMEM	O	IBoolean	R	-	G	Indicates if WRITEMEM_CMD and WRITEMEM_ACK are supported.
GevSupportedOptionalCommandsConcatenation	O	IBoolean	R	-	G	Indicates if multiple operations in a single message are supported.
GevHeartbeatTimeout	O	IInteger	R/W	ms	G	Current heartbeat timeout in milliseconds.
GevTimestampTickFrequency	O	IInteger	R	ticks	E	64-bit value indicating the number of timestamp clock tick in 1 second. Timestamp tick frequency is 0 if timestamp is not supported.
GevTimestampControlLatch	O	ICommand	W	-	E	Latches the current timestamp value of the device.
GevTimestampControlReset	O	ICommand	W	-	E	Resets the timestamp count of the device.
GevTimestampValue	O	IInteger	R	ticks	E	Latched 64-bit value of the timestamp. Value must first be latched using GevTimestampControlLatch.
GevCCP	O	IEnumeration	R/W	-	G	Control Channel Privilege feature. {ExclusiveAccess, ControlAccess}
GevMCPHostPort	O	IInteger	R/W	-	G	The port to which the device must send messages. Setting this value to 0 closes

						the message channel.
GevMCDA	O	IInteger	R/W	-	G	Message channel destination IPv4 address. The destination address can be a multicast or a unicast address.
GevMCTT	O	IInteger	R/W	ms	G	Message Channel Transmission Timeout in ms.
GevMCRC	O	IInteger	R/W	-	G	Number of retransmissions allowed on the message channel.
GevStreamChannelSelector	O	IInteger	R/W	-	G	Indicate which stream channel to configure.
GevSCPInterfaceIndex[GevStreamChannelSelector]	R	IInteger	R/W	-	G	Index of network interface to use (from 0 to 3). Specific streams might be hard-coded to a specific network interfaces. Therefore this field might not be programmable on certain devices. It is read-only for this case. Applies to the selected stream channel
GevSCPHostPort[GevStreamChannelSelector]	O	IInteger	R/W	-	G	The port to which the device must send data stream. Setting this value to 0 closes the stream channel. Applies to the selected stream channel.
GevSCPSFireTestPacket[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	When this bit is set, the device will fire one test packet of size specified by bit 0-15. The “don’t fragment” bit of IP header must be set for this test packet. Applies to the selected stream channel.
GevSCPSDoNotFragment[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	This bit is copied into the “do not fragment” bit of IP header of each stream packet. It can be used by the application to prevent IP fragmentation of packets on the stream channel. Applies to the selected stream channel.

GevSCPSBigEndian[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Endianess of multi-byte pixel data for this stream. FALSE: little endian TRUE: big endian This is an optional feature. A device that does not support this feature must support little-endian and always leave that bit clear. Applies to the selected stream channel.
GevSCPSPacketSize[GevStreamChannelSelector]	R	IInteger	R/W	-	E	The stream packet size to send on this channel, except for data leader and data trailer; and the last data packet which might be of smaller size (since packet size is not necessarily a multiple of block size for stream channel). The value is in bytes. If a device cannot support the requested packet size, then it must not fire a test packet when requested to do so. Applies to the selected stream channel.
GevSCPD[GevStreamChannelSelector]	R	IInteger	R/W	ticks	E	Inter-packet delay in timestamp tick. Applies to the selected stream channel.
GevSCDA[GevStreamChannelSelector]	O	IInteger	R/W	-	G	Stream channel destination IPv4 address. The destination address can be a multicast or a unicast address. Applies to the selected stream channel.
GevLinkSpeed	O	IInteger	R	Mbps	E	Connection speed in Mbps of the network interface selected by GevInterfaceSelector. Typically 10 or 100 or 1000 Mbps.
GevIPConfigurationStatus	O	IEnumeration	R	-	B	Reports the current IP configuration status.

2.10 User Sets

User Sets provides the features used to save camera settings to on-board non-volatile memory.

Table 2-10: User Sets Summary

Name	Level	Interface	Access	Unit	Visibility	Description
UserSetSelector	R	IEnumeration	R/W	-	B	Selects the feature User Set to load, save or configure. {Default, UserSet1, UserSet2, …}
UserSetLoad[UserSetSelector]	R	ICommand	W	-	B	Loads the User Set specified by UserSetSelector to the device and makes it active.
UserSetSave[UserSetSelector]	R	ICommand	W	-	B	Saves the selected User Set specified by UserSetSelector to persistent memory.
UserSetDefaultSelector	O	IEnumeration	R/W	-	B	Selects the feature User set to load at reset. {Default, UserSet1, UserSet2, …}

2.11 Chunk Data Streams

Chunk Data Streams provides feature to append information to image data.

Table 2-11: Chunk Data Streams Summary

Name	Level	Interface	Access	Unit	Visibility	Description
ChunkModeActive	R	IBoolean	R/W	-	E	Enables Chunk mode.
ChunkSelector	R	IEnumeration	R/W	-	E	Selects which Chunk to configure. {Image, OffsetX, OffsetY, Width, Height, Timestamp, LineStatusAll}
ChunkEnable[ChunkSelector]	R	IBoolean	R/W	-	E	Enables the inclusion of the selected Chunk in the payload data.
ChunkImage	R	IRegister	R	-	G	Entire pixel data of the captured image.
ChunkOffsetX	R	IInteger	R	pixels	E	Left coordinate of AOI.
ChunkOffsetY	R	IInteger	R	pixels	E	Top coordinate of AOI.
ChunkWidth	R	IInteger	R	pixels	E	Width of AOI.
ChunkHeight	R	IInteger	R	pixels	E	Height of AOI.
ChunkPixelFormat	R	IEnumeration	R	-	E	Format for the image pixels. {Mono8, Mono8Signed, Mono10, Mono10Packed, Mono12, Mono12Packed, Mono16, BayerGR8, BayerRG8, BayerGB8, BayerBG8, BayerGR10, BayerRG10, BayerGB10, BayerBG10, BayerGR12, BayerRG12, BayerGB12, BayerBG12, RGB8Packed, BGR8Packed, RGBA8Packed, BGRA8Packed, RGB10Packed, BGR10Packed, RGB12Packed, BGR12Packed, RGB10V1Packed, RGB10V2Packed, YUV411Packed, YUV422Packed, YUV444Packed, RGB8Planar, RGB10Planar, RGB12Planar, RGB16Planar}
ChunkDynamicRangeMax	R	IInteger	R	bits/pixel	E	Maximum value of the image.

ChunkDynamicRangeMin	R	IInteger	R	bits/pixel	E	Minimum value of the image.
ChunkTimestamp	R	IInteger	R	-	E	Time stamp at FrameStart time.
ChunkLineStatusAll	R	IInteger	R	bitfield	E	State of the input and output Pin as returned by LineStatusAll feature at FrameStart time.
ChunkCounterSelector	R	IEnumeration	R/W	-	E	Selects the Counter to read with ChunkCounter. {Counter1, Counter2, …}
ChunkCounter[ChunkCounterSelector]	R	IInteger	R	-	E	Value of the selected Chunk counter at the time of the FrameStart internal event.
ChunkTimerSelector	R	IEnumeration	R/W	-	E	Selects the Timer to read with ChunkTimer. {Timer1, Timer2, …}
ChunkTimer[ChunkTimerSelector]	R	IFloat	R	us	E	Value of the selected Timer at the time of the FrameStart internal event.

2.12 File Access

The File Access features provide all the sevices necessary for generic file access of a device.

Table 2-122: File Access controls Summary

Name	Level	Interface	Access	Unit	Visibility	Description
FileSelector (FS)	R	IEnumeration	R/(W)	-	G	Selector for the file to be accessed. {UserSetDefault, UserSet1, UserSet2, LUTLuminance, LUTRed, LUTGreen, LUTBlue, …}.
FileOperationSelector[FS]	R	IEnumeration	R/W	-	G	Selector for the operation {Open, Close, Read, Write, …}.
FileOperationExecute[FS][FOS]	R	ICommand	W	-	G	Executes the selected File Operation.
FileOpenModeSelector[FS] [FOS]	R	IEnumeration	R/W	-	G	Selects the access mode to use when doing an Open operation on the file. {Read, Write, ReadWrite, …}.
FileAccessOffset[FS][FOS]	R	IInteger	R/W	Byte	G	Offset position in the file to use for Read/Write access.
FileAccessLength[FS][FOS]	R	IInteger	R/W	Byte	G	Number of Bytes to use for Read/Write access.
FileAccessBuffer[FS][FOS]	R	IRegister	R/W	-	G	Access Buffer to exchange data between device File and the application. The effective buffer access start position and access length relative to the device file is specified with FileAccessOffset and FileAccessLength.
FileOperationStatus[FS][FOS]	R	IEnumeration	R	-	G	Status of the open/close operations. This feature must returns Success if the operation was executed exactly as specified. {Success, Failure, …}
FileOperationResult[FS][FOS]	R	IInteger	R	Byte	G	Returns the number of bytes successfully Read or Written during the last Operation.
FileSize[FS]	R	IInteger	R	Byte	G	Returns the size of the file.

3 Device Information

Device Information features provides general information about the camera and its sensor. This is mainly used to identify the camera during the enumeration process and to obtain information about the sensor resolution. Other information and controls pertaining to the general state of the camera are also included in this category.

3.1 DeviceVendorName

Name	DeviceVendorName
Level	Recommended
Interface	IString
Access	Read-only
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

This feature provides the name of the manufacturer of the device.

For GigE Vision bootstrap registers, this string has a maximum length of 32 bytes (including the NULL-terminating character).

3.2 DeviceModelName

Name	DeviceModelName
Level	Recommended
Interface	IString
Access	Read-only
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

This feature provides the model of the device.

For GigE Vision bootstrap registers, this string has a maximum length of 32 bytes (including the NULL-terminating character).

3.3 DeviceManufacturerInfo

Name	DeviceManufacturerInfo
Level	Recommended
Interface	IString
Access	Read-only
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

This feature provides extended manufacturer information about the device.

For GigE Vision bootstrap registers, this string has a maximum length of 48 bytes (including the NULL-terminating character).

3.4 DeviceVersion

Name	DeviceVersion
Level	Recommended
Interface	IString
Access	Read-only
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

This feature provides the version of the device.

For GigE Vision bootstrap registers, this string has a maximum length of 32 bytes (including the NULL-terminating character).

3.5 DeviceFirmwareVersion

Name	DeviceFirmwareVersion
Level	Recommended
Interface	IString
Access	Read-only

Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

This feature provides the version of the firmware in the device. This information is not provided by the standard GigE Vision bootstrap registers. It must be part of device-specific registers.

3.6 DeviceID

Name	DeviceID
Level	Recommended
Interface	IString
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	Any NULL-terminated string

This feature stores a camera identifier. This is typically the serial number of the device.

GigE Vision bootstrap registers provide a string with up to 16 bytes to store the serial number of the camera.

3.7 DeviceUserID

Name	DeviceUserID
Level	Optional
Interface	IString
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Any NULL-terminated string

This feature stores a user-programmable identifier. For GigE Vision bootstrap registers, this string has a maximum length of 16 bytes (including the NULL-terminating character).

When this feature is present, it must be writable.

3.8 DeviceScanType

Name	DeviceScanType
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	Areascan Linescan

This feature specifies the scan type of the sensor. Typically, this feature is not writable. But some cameras might allow switching between linescan and areascan.

DeviceScanType can take any of the following values:

- **Areascan**: 2D sensor
- **Linescan**: 1D sensor

3.9 DeviceMaxThroughput

Name	DeviceMaxThroughput
Level	Optional
Interface	IInteger
Access	Read
Unit	bytes/sec
Recommended Visibility	Expert
Values	>0

This feature indicates the maximum bandwidth of data that can be streamed out of the device. This can be used to estimate if the network connection can sustain transfer of free-running images from the camera at its maximum speed.

3.10 DeviceReset

Name	DeviceReset
Level	Recommended
Interface	ICommand
Access	Write-only
Unit	-
Recommended Visibility	Guru
Values	-

This command is used to reset the device and to put it in its power up state.

3.11 DeviceRegistersStreamingStart

Name	DeviceRegistersStreamingStart
Level	Recommended
Interface	ICommand
Access	Write-only
Unit	-
Recommended Visibility	Guru
Values	-

This command is used to prepare for registers streaming without immediate checking for consistency (blind registers setting). If the camera implements this feature, GenApi guarantees using it to announce register streaming.

3.12 DeviceRegistersStreamingEnd

Name	DeviceRegistersStreamingEnd
Level	Recommended
Interface	ICommand
Access	Write-only
Unit	-
Recommended Visibility	Guru

Values	-

This command is used to announce the end of registers streaming. This will do a register set validation for consistency and activate it. This will also update the **DeviceRegistersValid** flag.

3.13 DeviceRegistersCheck

Name	DeviceRegistersCheck
Level	Recommended
Interface	ICommand
Access	Write-only
Unit	-
Recommended Visibility	Expert
Values	-

This command is used to perform the validation of the current register set for consistency. This will update the **DeviceRegistersValid** flag.

3.14 DeviceRegistersValid

Name	DeviceRegistersValid
Level	Recommended
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	True False

This feature is used to read if the current register set is valid and consistent.

4 Image Size Control

This section describes how to influence and determine the image size and format. It also provides the necessary information to acquire and to display the image data. It assumes that the camera expels a single rectangular image.

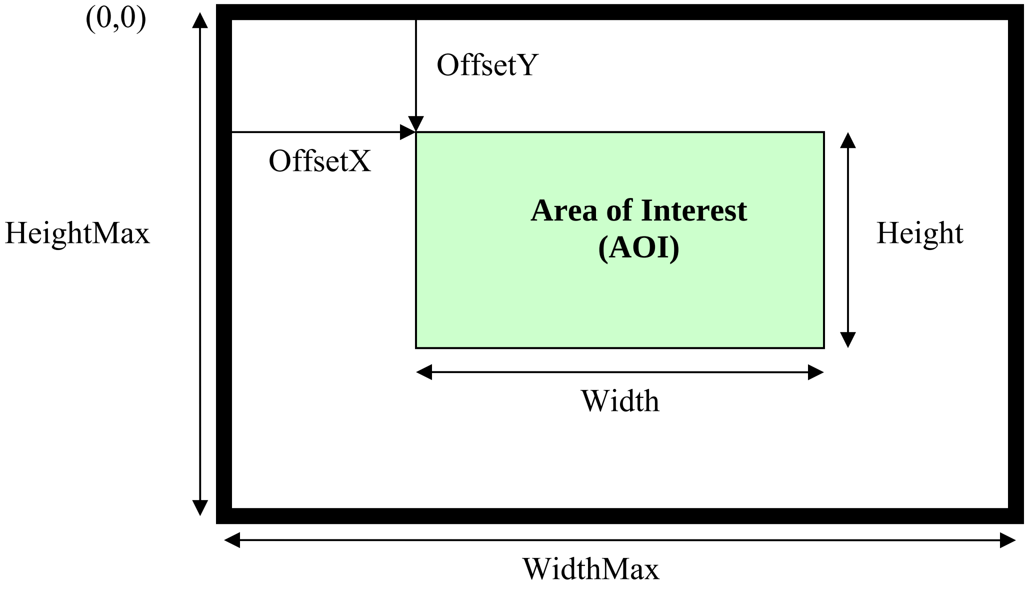

Figure 4-1: Image size and defining an area of interest (AOI)

The sensor provides **SensorWidth** time **SensorHeight** pixels.

Using **BinningHorizontal** and/or **BinningVertical** or **DecimationHorizontal** and/or **DecimationVertical** the image is shrunk to**WidthMax** time **HeightMax** pixels.

In addition the features **ReverseX** and **ReverseY** can be used to flip the image respectively along the X-axis or Y-axis. The flipping is done before the AOI is applied.

Within the shrunk image the user can set an area of interest (AOI) using the features **OffsetX**, **OffsetY**, **Width**, and **Height**. The resulting image expelled by the camera has **Width** time **Height** pixels. **OffsetX** and **OffsetY** are given with respect to the upper left corner of the image which has the coordinate (0, 0), see Figure 4-1.

All measures are given in the unit [pixel]. As a result the values should not change if the **PixelFormat** changes. For monochrome cameras each pixel corresponds to one gray value. For color camera in raw mode (Bayer pattern, etc.) each pixel corresponds to one pixel in the color mask. For color cameras in RGB mode each pixel corresponds to one RGB triplet. For color cameras in YUV mode each pixel corresponds to one Y value with the associated color information.

The feature **Height** describes the height of the image in lines. The pixels within a line are contiguous. The lines however may be not contiguous, e.g. in order to yield a DWORD alignment. **LinePitch** gives the number of bytes separating the starting pixels of two consecutive lines.

Each pixel in the image has a format defined by **PixelFormat**. For details see GigEVision specification **PixelFormat** (section 25.2 of GigE Vision Specification).

Because the **PixelFormat** feature contains a mix of informations specified by the user and informations provided by the camera, it is suitable for describing the whole pixel settings but might be less practical when individual setting must be set or inquired. Therefore a second set of features exists composed of the individual components of **PixelFormat.** Those features are **PixelCoding**, **PixelSize**, **PixelColorFilter**, **PixelDynamicRangeMin** and **PixelDynamicRangeMax.**

Even if the **PixelFormat** might allow for, e.g. 16 bits per pixel, the real image data might provide only a certain range of value (e.g. 12 bits per pixel because the camera is equipped with a 12 bit analog to digital converter only). In that case, **DynamicRangeMin** and **DynamicRangeMax** specify the lower and upper limits of the pixel values in the image. In general, **DynamicRangeMin** should be zero and **DynamicRangeMax** should be a power of two ($\left[0, 2^{DataDepth} - 1\right]$). There should be no missing codes in the range.

4.1 SensorWidth

Name	SensorWidth
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	Pixels
Recommended Visibility	Expert
Values	>0

This feature indicates the effective width of the sensor in pixels. Its value must be greater than 0.

4.2 SensorHeight

Name	SensorHeight
Level	Recommended

Interface	IInteger
Access	Read-only
Unit	Pixels
Recommended Visibility	Expert
Values	>0

This feature indicates the effective height of the sensor in pixels. Its value must be greater than 0. For linescan sensor, this value is 1.

4.3 SensorTaps

Name	SensorTaps
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	One Two Three Four Device-specific

This feature represents the number of taps of the camera sensor.

4.4 SensorDigitizationTaps

Name	SensorDigitizationTaps
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert

Values	One Two Three Four Device-specific

This feature represents the number of digitized samples outputted simultaneously by the camera A/D conversion stage.

4.5 WidthMax

Name	WidthMax
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	Pixels
Recommended Visibility	Expert
Values	>0

This feature represents the maximum width (in pixels) of the image after horizontal binning, decimation or any other function changing the horizontal dimensions of the image.

4.6 HeightMax

Name	HeightMax
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	Pixels
Recommended Visibility	Expert
Values	>0

This feature represents the maximum height (in pixels) of the image after vertical binning, decimation or any other function changing the vertical dimensions of the image.

4.7 Width

Name	Width
Level	Mandatory
Interface	IInteger
Access	Read/(Write)
Unit	pixels
Recommended Visibility	Beginner
Values	>0

This feature represents the actual image width expelled by the camera (in pixels).

4.8 Height

Name	Height
Level	Mandatory
Interface	IInteger
Access	Read/(Write)
Unit	pixels
Recommended Visibility	Beginner
Values	>0

This feature represents the actual image height expelled by the camera (in pixels).

4.9 OffsetX

Name	OffsetX
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	pixels
Recommended	Beginner

Visibility	
Values	≥0

This feature represents the horizontal offset from the origin to the AOI (in pixels).

4.10 OffsetY

Name	OffsetY
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	pixels
Recommended Visibility	Beginner
Values	≥0

This feature represents the vertical offset from the origin to the AOI (in pixels).

4.11 LinePitch

Name	LinePitch
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	bytes
Recommended Visibility	Expert
Values	≥0

This feature is used to facilitate alignment of image data. It indicates the total number of bytes between 2 successive lines. This might be useful if the system has specific limitations, such as having the lines aligned on 32-bit boundaries.

4.12 BinningHorizontal

Name	BinningHorizontal

Level	Optional
Interface	IInteger
Access	Read/Write
Unit	Cells
Recommended Visibility	Expert
Values	>0

This feature represents the number of horizontal photo-sensitive cells that must be combined (added) together. This has the net effect of increasing the intensity (or signal to noise ratio) of the pixel and reducing the horizontal resolution (width) of the image.

A value of 1 indicates that no horizontal binning is performed by the camera.

4.13 BinningVertical

Name	BinningVertical
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	cells
Recommended Visibility	Expert
Values	>0

This feature represents the number of vertical photo-sensitive cells that must be combined (added) together. This has the net effect of increasing the intensity (or signal to noise ratio) of the pixel and reducing the vertical resolution (height) of the image.

A value of 1 indicates that no vertical binning is performed by the camera.

4.14 DecimationHorizontal

Name	DecimationHorizontal
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	cells

Recommended Visibility	Expert
Values	≥0

This feature allows horizontal sub-sampling of the image. This might be realized by pixel dropping or by first applying a horizontal low-pass filter before dropping pixels. This has the net effect of reducing the horizontal resolution (width) of the image by the specified horizontal decimation factor.

A value of 1 indicates that the camera performs no horizontal decimation.

4.15 DecimationVertical

Name	DecimationVertical
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	cells
Recommended Visibility	Expert
Values	≥0

This feature allows vertical sub-sampling of the image. This might be realized by pixel dropping or by first applying a vertical low-pass filter before dropping pixels. This has the net effect of reducing the vertical resolution (height) of the image by the specified vertical decimation factor.

A value of 1 indicates that the camera performs no vertical decimation.

4.16 ReverseX

Name	ReverseX
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

This feature is used to flip horizontally the image sent by the device. The AOI is applied after the flipping.

4.17 ReverseY

Name	ReverseY
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

This feature is used to flip vertically the image sent by the device. The AOI is applied after the flipping.

4.18 PixelFormat

Name	PixelFormat
Level	Mandatory
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Beginner
Values	Mono8 Mono8Signed Mono10 Mono10Packed Mono12 Mono12Packed Mono16 BayerGR8 BayerRG8 BayerGB8 BayerBG8

	BayerGR10 BayerRG10 BayerGB10 BayerBG10 BayerGR12 BayerRG12 BayerGB12 BayerBG12 RGB8Packed BGR8Packed RGBA8Packed BGRA8Packed RGB10Packed BGR10Packed RGB12Packed BGR12Packed RGB10V1Packed RGB10V2Packed YUV411Packed YUV422Packed YUV444Packed RGB8Planar RGB10Planar RGB12Planar RGB16Planar Device-specific

This feature indicates the format of the pixel to use during the acquisition. Values of the enumeration and the pixel formatting correspond to the GigE Vision specification. It contains all the informations provided by **PixelCoding, PixelSize**, **PixelColorFilter** but combined in one single value.

4.19 PixelCoding

Name	PixelCoding
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert

Values	Mono MonoSigned MonoPacked RGBPacked BGRPacked RGBAPacked BGRAPacked RGBPlanar YUV411Packed YUV422Packed YUV444Packed Raw

This feature indicates the coding of the pixels in the image. Raw gives the data in the native fomat of the sensor. It is mainly used for Bayer sensor. This value must always be coherent with the **PixelFormat** feature.

4.20 PixelSize

Name	PixelSize
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	Bpp8 Bpp10 Bpp12 Bpp14 Bpp16 Bpp24 Bpp32 Bpp64

This feature indicates the total size in bits of a pixel of the image. This value must always be coherent with the **PixelFormat** feature.

4.21 PixelColorFilter

Name	PixelColorFilter

Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Expert
Values	None BayerRG BayerGB BayerGR BayerBG

This feature indicates the type of color filter that is applied to the image. This value must always be coherent with the **PixelFormat** feature.

4.22 PixelDynamicRangeMin

Name	PixelDynamicRangeMin
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Device-specific

This feature indicates the minimal value that can be returned during the digitization process. This corresponds to the darkest value of the camera. For color camera, this returns the smallest value that each color component can take.

4.23 PixelDynamicRangeMax

Name	PixelDynamicRangeMax
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-

Recommended Visibility	Expert
Values	Device-specific

This feature indicates the maximal value that will be returned during the digitization process. This corresponds to the brightest value of the camera. For color camera, this returns the biggest value that each color component can take.

4.24 TestImageSelector

Name	TestImageSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off Black White GreyHorizontalRamp GreyVerticalRamp GreyHorizontalRampMoving GreyVerticalRampMoving HorzontalLineMoving VerticalLineMoving ColorBar FrameCounter Device-specific

This feature selects the type of test image that is expelled by the camera.

TestImageSelector can take any of the following values:

- **Off**: Image is coming from the sensor.
- **Black**: Image is filled with the darkest possible image.
- **White**: Image is filled with the brightest possible image.
- **GreyHorizontalRamp**: Image is filled horizontally with an image that goes from the darkest possible value to the brightest.

- **GreyVerticalRamp**: Image is filled vertically with an image that goes from the darkest possible value to the brightest.
- **GreyHorizontalRampMoving**: Image is filled horizontally with an image that goes from the darkest possible value to the brightest and that moves horizontally from left to right at each frame.
- **GreyVerticalRamp**: Image is filled vertically with an image that goes from the darkest possible value to the brightest and that moves verticaly from top to bottom at each frame.
- **HorizontalLineMoving**: A moving horizontal line is superimposed on the live image.
- **VerticalLineMoving**: A moving vertical line is superimposed on the live image.
- **ColorBar**: Image is filled with stripes of color including White, Black, Red, Green, Blue, Cyan, Magenta and Yellow.
- **FrameCounter**: A frame counter is superimposed on the live image.

Other values are device-specific and represent particular test images digitally generated by the camera.

5 Acquisition and Trigger Controls

The Acquisition and Trigger Controls section describes all features related to image acquisition, including the trigger and exposure control. It describes the basic model for acquisition and the typical behavior of the device.

An **Acquisition** is defined as the capture of a sequence of one or many **Frame**(s) (see Figure 5-1**).** The transfer of the frame(s) of an **Acquisition,** starts with the beginning of the transfer of the first frame and ends with completion of the transfer of the last one.

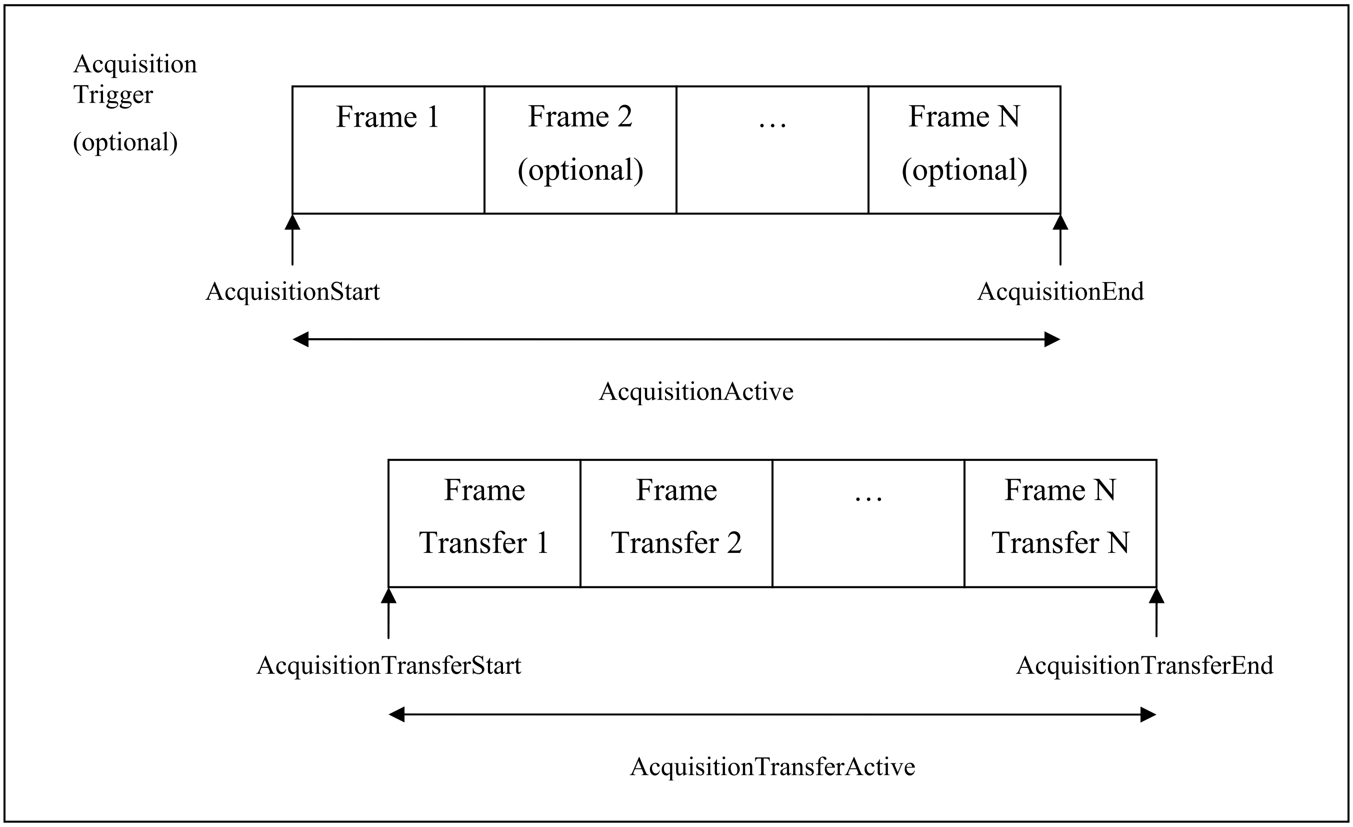

Figure 5-1: Acquisition signals definitions

A **Frame** is defined as the capture of **Width** pixels x **Height** lines. A **Frame** starts with an optional **Exposure** period and ends with the completion of the sensor read out. Generally, a transfer period will start during the sensor read out and will finish sometime after it but it is not considered as part of the Frame (see Figure 5-2).

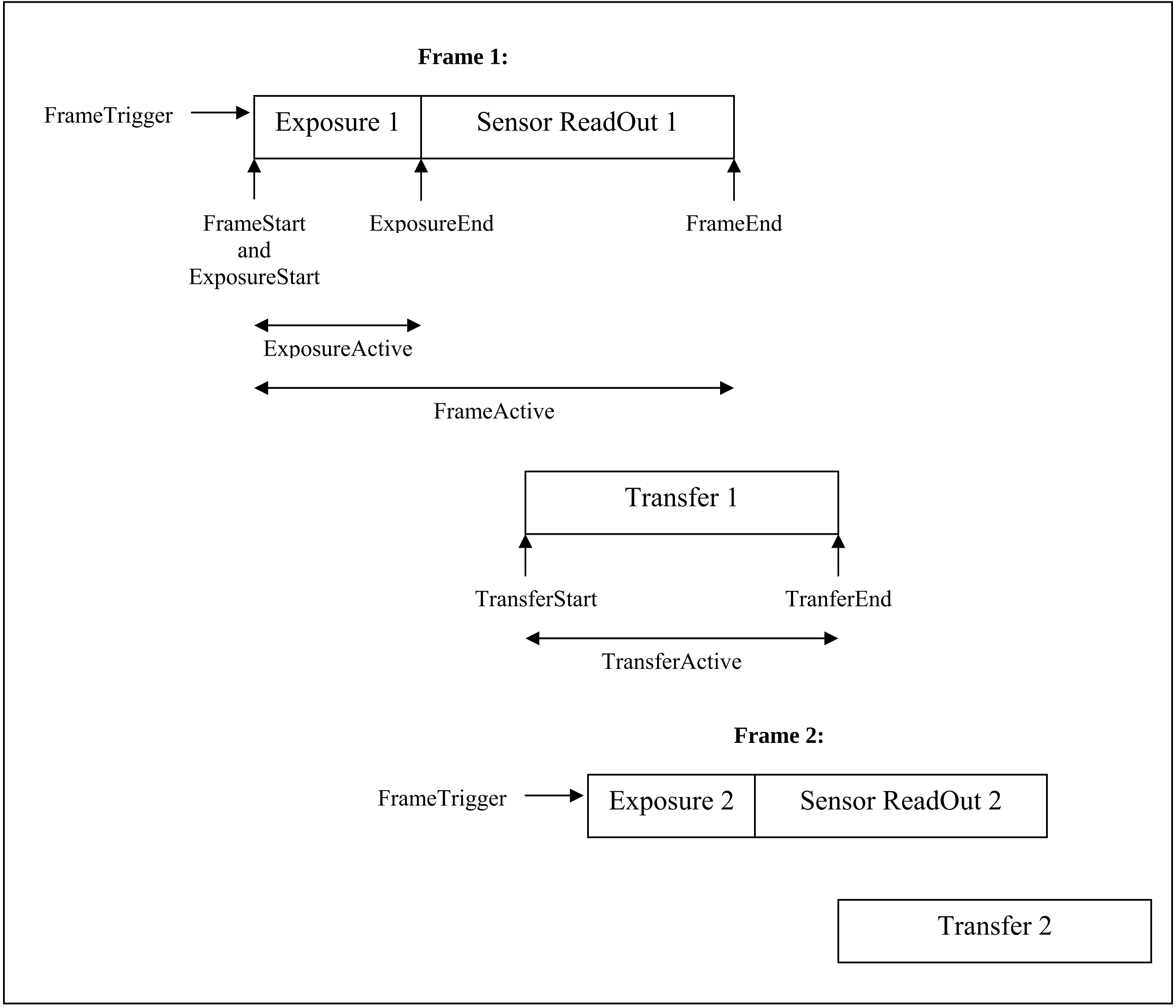

Figure 5-2: Frame signals definitions

For Line Scan acquisition, the definition of **Frame** stays the same but the exposure and read out are done for each line of the virtual Frame (see Figure 5-3).

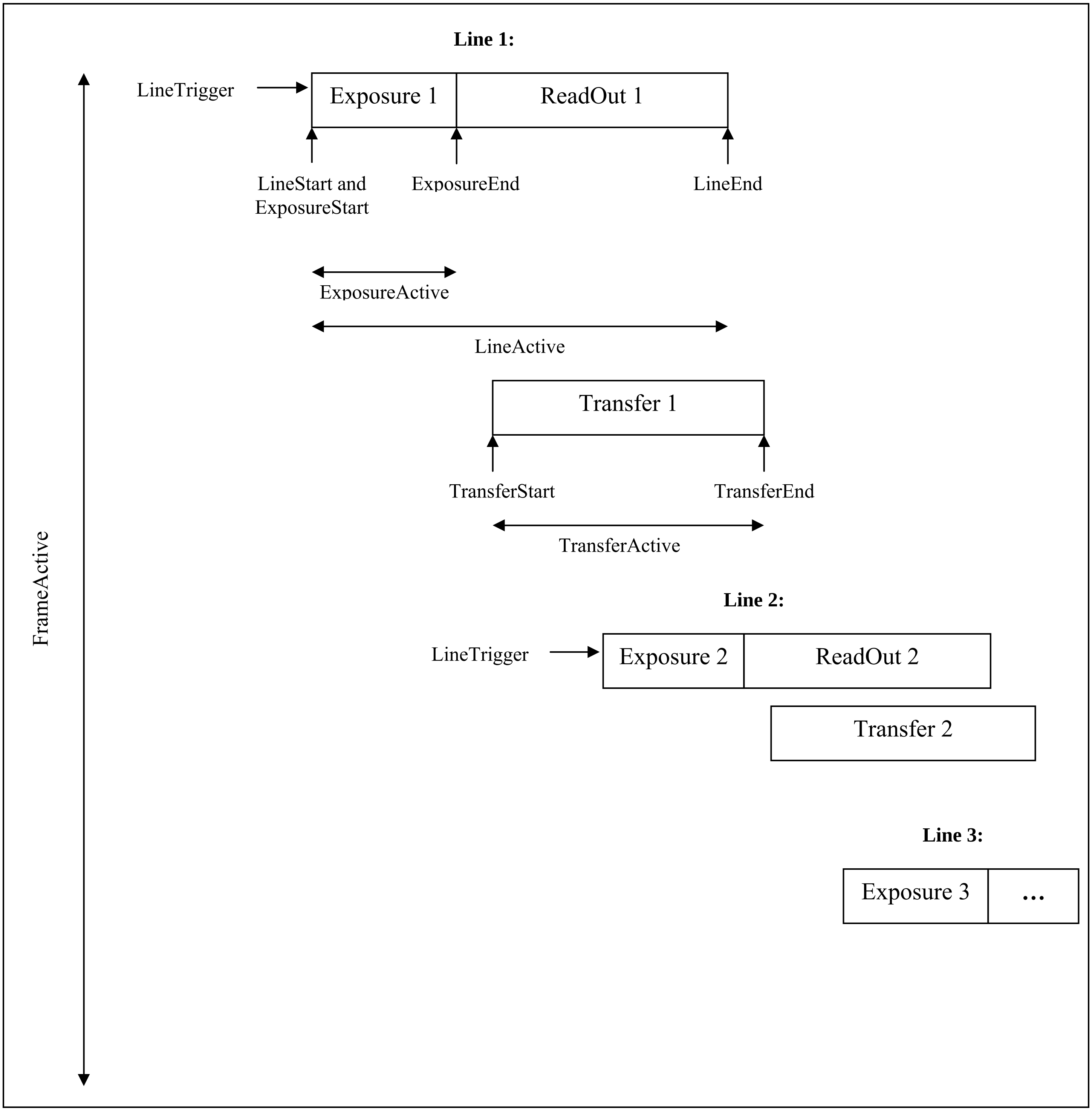

Figure 5-3: Frame signals definitions in Line Scan mode

Acquisition Control features:

The **AcquisitionMode** controls the mode of acquisition for the device. This mainly affects the number of frames captured in the Acquisition (**SingleFrame, MultiFrame, Continuous**).

The optional **AcquisitionArm** command is used to verify and freeze all parameters relevant for the image data capture. It prepares the device for the **AcquistionStart**.

The **AcquisitionStart** command is used to start the Acquisition.

The **AcquisitionStop** command will stop the Acquisition at the end of the current Frame**.** It can be used in any acquisition mode and if the camera is waiting for a trigger, the pending Frame will be cancelled.

The **AcquisitionAbort** command can be used to abort an Acquisition at any time. This will end the capture immediately without completing the current Frame.

AcquisitionFrameCount controls the number of frames that will be captured when **AcquisitionMode** is **MultiFrame**.

AcquisitionFrameRateAbs or **AcquisitionFrameRateRaw** controls the rate at which the Frames are captured when **TriggerMode** is **Off**.

AcquisitionLineRateAbs or **AcquisitionLineRateRaw** controls the rate at which the Lines in each Frame are captured. This is generally useful for line scan cameras.

AcquisitionStatusSelector and **AcquisitionStatus** can be used to read the status of the internal acquisition signals. The standard acquisition signals Status are: **AcquisitionTriggerWait, AcquisitionActive, AcquisitionTransfer, FrameTriggerWait, FrameActive, FrameTransfer, ExposureActive** (see Figure 5-1 and Figure 5-2),

See Chapter 14: **Typical Acquisition and Trigger examples** for more complete use cases of the acquisition and trigger features in conjunction with other related sections such as I/O and analog controls.

5.1 AcquisitionMode

Name	AcquisitionMode
Level	Mandatory
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	SingleFrame

	MultiFrame Continuous

This feature controls the acquisition mode of the device. It defines mainly the number of frames to capture during an acquisition and the way the acquisition stops. **AcquisitionMode** can take any of the following values:

- **SingleFrame**: One frame is captured.
- **MultiFrame**: The number of frames specified by **AcquisitionFrameCount** is captured.
- **Continuous**: Frames are captured continuously until stopped with the **AcquisitionStop** command.

5.2 AcquisitionStart

Name	AcquisitionStart
Level	Mandatory
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Beginner
Values	-

This feature starts the Acquisition of the device. The number of frames captured is specified by **AcquisitionMode.**

Note that unless the **AcquisitionArm** was executed since the last feature change, the **AcquisitionStart** command must validate all the current features for consistency before starting the Acquisition. This validation will not be repeated for the subsequent acquisitions unless a feature is changed in the device.

5.3 AcquisitionStop

Name	AcquisitionStop
Level	Mandatory
Interface	ICommand
Access	Write
Unit	-

Recommended Visibility	Beginner
Values	-

This feature stops the Acquisition of the device at the end of the current Frame**.** It is mainly used when **AcquisitionMode** is **Continuous** but can be used in any acquisition mode and if the camera is waiting for a trigger, the pending Frame will be cancelled. If no Acquisition is in progress, the command is ignored.

5.4 AcquisitionAbort

Name	AcquisitionAbort
Level	Recommended
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Expert
Values	-

This feature aborts an Acquisition at any time. This will end the capture immediately without completing the current Frame or waiting on a trigger. If no Acquisition is in progress, the command is ignored.

5.5 AcquisitionArm

Name	AcquisitionArm
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Expert
Values	-

This feature can be used to arm the device before an **AcquisitionStart** command. This optional command validates all the current features for consistency and prepares the device for a fast start of the Acquisition. If not used explicitly, this command will be automatically executed at the first **AcquisitionStart** but will not be repeated for the subsequent ones unless a feature is changed in the device.

5.6 AcquisitionFrameCount

Name	AcquisitionFrameCount
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	Frames
Recommended Visibility	Beginner
Values	≥1

This feature provides the number of frames to be acquired in MultiFrame Acquisition mode. The minimum allowable value is 1.

5.7 AcquisitionFrameRateAbs

Name	AcquisitionFrameRateAbs
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	Hz
Recommended Visibility	Beginner
Values	Device-specific

This feature controls the rate (in Hertz) at which the Frames are captured when **TriggerMode** is **Off** for the Frame trigger.

5.8 AcquisitionFrameRateRaw

Name	AcquisitionFrameRateRaw
Level	Optional
Interface	IInteger

Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Device-specific

This feature controls the rate (in device specific unit) at which the Frames are captured when **TriggerMode** is **Off** for the Frame trigger.

5.9 AcquisitionLineRateAbs

Name	AcquisitionLineRateAbs
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	Hz
Recommended Visibility	Beginner
Values	Device-specific

This feature controls the rate (in Hertz) at which the Lines in a Frame are captured when **TriggerMode** is **Off** for the Line trigger. This is generally useful for line scan camera only.

5.10 AcquisitionLineRateRaw

Name	AcquisitionLineRateRaw
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Device-specific

This feature controls the rate (in device specific unit) at which the Lines in a Frame are captured when **TriggerMode** is **Off** for the Line trigger. This is generally useful for line scan camera only.

5.11 AcquisitionStatusSelector

Name	AcquisitionStatusSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	AcquisitionTriggerWait AcquisitionActive AcquisitionTransfer FrameTriggerWait FrameActive FrameTransfer ExposureActive

This feature is used to select which internal acquisition signal to read using AcquisitionStatus**.**

AcquisitionStatusSelector can take any of the following values (see Figure 5-1 and Figure 5-2):

- **AcquisitionTriggerWait**: Device is currently waiting for a trigger for the capture of one or many frames.
- **AcquisitionActive**: Device is currently doing an acquisition of one or many frames.
- **AcquisitionTransfer**: Device is currently transferring an acquisition of one or many frames.
- **FrameTriggerWait**: Device is currently waiting for a Frame trigger.
- **FrameActive**: Device is currently doing the capture of a frame.
- **FrameTransfer**: Device is currently transferring a frame.
- **ExposureActive**: Device is doing the Exposure of a frame.

5.12 AcquisitionStatus

Name	AcquisitionStatus[AcquisitionStatusSelector]
Level	Recommended
Interface	IBoolean
Access	Read
Unit	-

Recommended Visibility	Expert
Values	True False

This feature is used to read the state (True or False) of the internal acquisition signal selected using **AcquisitionStatusSelector.**

Trigger Control features:

The Trigger Controls section describes all features related to image acquisition using trigger(s).

One or many **Trigger**(s) can be used to control the start of an **Acquisition** (see Figure 5-1)**,** of a **Frame** of an Acquisition (see Figure 5-2) or each **Line** of a Frame (for line scan devices)**.** It can also be used to control the exposure duration at the beginning of a frame.

TriggerSelector is used to select which type of trigger to configure. The standard trigger types are: **AcquisitionStart, AcquisitionEnd, AcquisitionActive, FrameStart, FrameEnd, FrameActive, LineStart, ExposureStart, ExposureEnd** and **ExposureActive.**

TriggerMode activate/desactivate trigger operation. It can be **Off** or **On.**

TriggerSource specifies the physical input **Line** or internal signal to use for the selected trigger. Standard trigger sources are: **Software, Line0, Line1, …, Timer1Start, Timer1End, , …, Counter1Start, Counter1End, …, UserOutput0, UserOutput1, …**

With a **Software** trigger source, the **TriggerSoftware** command can be used by an application to generate an internal trigger signal.

With the hardware trigger sources, **TriggerActivation** specifies the activation mode of the trigger. This can be a **RisingEdge, FallingEdge, AnyEdge, LevelHigh** or **LevelLow**.

TriggerOverlap specifies the type of trigger overlap permitted with the previous frame. This defines when a valid trigger will be accepted (or latched) for a new frame. This can be **Off** for no overlap, **ReadOut** to accept a trigger immediately after the exposure period or **PreviousFrame** to accept (latch) a trigger that happened at any time after the start of the previous frame.

TriggerDelayAbs or **TriggerDelayRaw** specifies the delay to apply after the trigger reception before to effectively activate it.

TriggerDivider and **TriggerMultiplier** are used to control the ratio of triggers that are accepted.

For example to setup a hardware triggered acquisition that will start the capture of each frame on the rising edge of the signal coming from the physical input Line 1, the following pseudo-code can be used:

```
Camera.TriggerSelector     = FrameStart;
Camera.TriggerMode         = On;
Camera.TriggerActivation   = RisingEdge;
Camera.TriggerSource       = Line1;
```

See also Chapter 14: **Typical Acquisition and Trigger examples** for more complete use cases of the acquisition and trigger features in conjunction with other related sections such as I/O and analog controls.

5.13 TriggerSelector

Name	TriggerSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	AcquisitionStart AcquisitionEnd AcquisitionActive FrameStart FrameEnd FrameActive LineStart ExposureStart ExposureEnd ExposureActive

This feature is used to select which type of trigger to configure.

TriggerSelector can take any of the following values (see Figure 5-1 and Figure 5-2):

- **AcquisitionStart**: Selects a trigger that starts the Acquisition of one or many frames according to **AcquisitionMode**.
- **AcquisitionEnd**: Selects a trigger that ends the Acquisition of one or many frames according to **AcquisitionMode**.
- **AcquisitionActive**: Selects a trigger that controls the duration of the Acquisition of one or many frames.

- **FrameStart**: Selects a trigger starting the capture of one frame.
- **FrameEnd**: Selects a trigger ending the capture of one frame (mainly used in line scan mode).
- **FrameActive**: Selects a trigger controlling the duration of one frame (mainly used in line scan mode).
- **LineStart**: Selects a trigger starting the capture of one Line of a Frame (mainly used in line scan mode).
- **ExposureStart**: Selects a trigger controlling the start of the exposure of one Frame (or Line).
- **ExposureEnd**: Selects a trigger controlling the end of the exposure of one Frame (or Line).
- **ExposureActive**: Selects a trigger controlling the duration of the exposure of one frame (or Line).

5.14 TriggerMode

Name	TriggerMode[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off On

TriggerMode define if the selected trigger is active. It can take any of the following values:

- **Off**: Disables the selected trigger.
- **On**: Enable the selected trigger.

5.15 TriggerSoftware

Name	TriggerSoftware[TriggerSelector]
Level	Recommended
Interface	ICommand
Access	Write

Unit	-
Recommended Visibility	Beginner
Values	-

TriggerSoftware is a command that can be used by an application to generate an internal trigger when **TriggerSource** is set to **Software.**

5.16 TriggerSource

Name	TriggerSource[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Software, Line0, Line1, Line2, … Timer1Start, Timer2Start, … Timer1End, Timer2End, … Counter1Start, Counter2Start, … Counter1End, Counter2End, … UserOutput0, UserOutput1, …

TriggerSource specifies the internal signal or physical input **Line** to use as the trigger source for the selected trigger when **TriggerMode** is **On**. **TriggerSource** can take any of the following values:

- **Software**: Specifies that the trigger source will be generated by software using the **TriggerSoftware** command.
- **Line0, Line1, Line2,** …: Specifies which physical line (or pin) and associated I/O control block to use as external source for the trigger signal.
- **Timer1Start, Timer2Start, …, Timer1End, Timer2End,** …: Specifies which Timer signal to use as internal source for the trigger.
- **Counter1Start, Counter2Start, …, Counter1End, Counter2End,** …: Specifies which of the Counter signal to use as internal source for the trigger.
- **UserOutput0, UserOutput1, UserOutput2,** …: Specifies which User Output bit signal to use as internal source for the trigger.

5.17 TriggerActivation

Name	TriggerActivation[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

TriggerActivation specifies the activation mode of the trigger. **TriggerActivation** can take any of the following values:

- **RisingEdge**: Specifies that the trigger is considered valid on the rising edge of the source signal.
- **FallingEdge**: Specifies that the trigger is considered valid on the falling edge of the source signal.
- **AnyEdge**: Specifies that the trigger is considered valid on the falling or rising edge of the source signal.
- **LevelHigh**: Specifies that the trigger is considered valid as long as the level of the source signal is high.
- **LevelLow**: Specifies that the trigger is considered valid as long as the level of the source signal is low.

5.18 TriggerOverlap

Name	TriggerOverlap[TriggerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert

Values	Off ReadOut PreviousFrame

TriggerOverlap specifies the type trigger overlap permitted with the previous frame. This defines when a valid trigger will be accepted (or latched) for a new frame. It can take any of the following values:

- **Off**: No trigger overlap is permitted.
- **ReadOut**: Trigger is accepted immediately after the exposure period.
- **PreviousFrame**: Trigger is accepted (latched) at any time during the capture of the previous frame.

5.19 TriggerDelayAbs

Name	TriggerDelayAbs[TriggerSelector]
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Expert
Values	Device-specific

TriggerDelayAbs specifies the absolute delay in microseconds (us) to apply after the trigger reception before effectively activating it. **TriggerDelayRaw** must reflect the state of **TriggerDelayAbs** when they are both supported.

5.20 TriggerDelayRaw

Name	TriggerDelayRaw[TriggerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	Device-specific
Recommended Visibility	Expert

Values	Device-specific

TriggerDelayRaw specifies the delay in device-specific unit to apply after the trigger reception before effectively activating it. **TriggerDelayAbs** must reflect the state of **TriggerDelayRaw** when they are both supported**.**

5.21 TriggerDivider

Name	TriggerDivider[TriggerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	Trigger signal
Recommended Visibility	Expert
Values	Device-specific

TriggerDivider is used to divide the number of incoming trigger pulses by an integer factor.

5.22 TriggerMultiplier

Name	TriggerMultiplier[TriggerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	Trigger signal
Recommended Visibility	Expert
Values	Device-specific

TriggerMultiplier is used to multiply the number of incoming trigger pulses by an integer factor. It is used generally used in conjunction with **TriggerDivider** to control the ratio of triggers that are accepted.

<u>Exposure Control features:</u>

The Exposure Controls section describes all features related to the exposure of the photosensitive cells (shutter control) during image acquisition.

The Exposure of the photosensitive cells during Frame or Line acquisition can be in 3 differents modes.

- **ExposureMode** can be **Off** to disable the Shutter and let it open.
- **ExposureMode** can be **Timed** to have a timed exposure and allow programing the duration using the **ExposureTimeAbs, ExposureTimeRaw or ExposureTimeAuto** features.

  For example to have a fixed exposure time of 1 milisecond, use the following pseudo code:

```
Camera.ExposureMode    = Timed;
Camera.ExposureTimeAbs = 1000;
```

- **ExposureMode** can be **TriggerWidth** to use the width of the current Frame or Line trigger signal(s) to control exposure duration.
- **ExposureMode** can be **TriggerControlled** to use one or more trigger signal(s) to control the exposure duration independently from the current Frame or Line triggers (See **ExposureStart, ExposureEnd** and **ExposureActive** of the **TriggerSelector** feature).

  For example: To use 2 hardware triggers respectively starting and stopping the Exposure, use the following pseudo code:

```
Camera.ExposureMode    = TriggerControlled;
Camera.TriggerSelector = ExposureStart;
Camera.TriggerMode     = On;
Camera.TriggerSource   = Line1;
Camera.TriggerSelector = ExposureEnd;
Camera.TriggerMode     = On;
Camera.TriggerSource   = Line2;
```

5.23 ExposureMode

Name	ExposureMode
Level	Recommended
Interface	IEnumeration
Access	Read/Write

Unit	-
Recommended Visibility	Beginner
Values	Off Timed TriggerWidth TriggerControlled

This feature is used to set the operation mode of the Exposure (or shutter).

ExposureMode can take any of the following values:

- **Off**: Disables the Exposure and let the shutter open.
- **Timed**: Timed exposure. The exposure duration time is set using the **ExposureTimeAbs, ExposureTimeRaw** or **ExposureTimeAuto** features and the exposure starts with the FrameStart (see Figure 5-2).
- **TriggerWidth**: Uses the width of the current Frame or Line trigger signal(s) pulse to control the exposure duration. Note that if the Frame or Line **TriggerActivation** is RisingEdge or LevelHigh, the exposure duration will be the time the trigger stays High. If **TriggerActivation** is FallingEdge or LevelLow, the exposure time will last as long as the trigger stays Low.
- **TriggerControlled**: Uses one or more trigger signal(s) to control the exposure duration independently from the current Frame or Line triggers. See **ExposureStart, ExposureEnd** and **ExposureActive** of the **TriggerSelector** feature.

Note also that **ExposureMode** as priority over the Exposure Trigger settings defined using **TriggerSelector=Exposure…** and defines which trigger (if any) is active.

For example, if:

```
ExposureMode    = Timed;
ExposureTimeAbs = 200;
```

Then the Exposure will be controlled using the **ExposureTimeAbs** Feature, even if the following code is done:

```
TriggerSelector   = ExposureActive;
TriggerMode       = On;
TriggerActivation = LevelHigh;
TriggerSource     = Line1;
```

But simply by adding:

```
ExposureMode = TriggerControlled;
```

The Exposure duration will become controlled by the length of the positive pulse on physical Line 1.

5.24 ExposureTimeAbs

Name	ExposureTimeAbs
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Beginner
Values	≥0

This feature is used to set the Exposure time (in microseconds) when **ExposureMode** is **Timed**. This controls the duration where the photosensitive cells are exposed to light.

ExposureTimeRaw must reflect the value of **ExposureTimeAbs** when they are both supported.

5.25 ExposureTimeRaw

Name	ExposureTimeRaw
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	Device-specific
Recommended Visibility	Beginner
Values	≥0

This feature is used to set the Exposure time in device-specific unit when **ExposureMode** is **Timed**. This controls the duration where the photosensitive cells are exposed to light.

ExposureTimeAbs must reflect the value of **ExposureTimeRaw** when they are both supported.

5.26 ExposureAuto

Name	ExposureAuto
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off Once Continuous Device-specific

This feature performs automatic exposure control when **ExposureMode** is **Timed**. The exact algorithm used to implement this control is device-specific. Some other device-specific features might be used to allow the selection of the algorithm.

ExposureAuto can take any of the following values:

- **Off**: Exposure duration is manually controlled using **ExposureTimeAbs** and **ExposureTimeRaw.**
- **Once**: Exposure duration is adapted once by the device. Once it has converged, it returns to the **Off** state.
- **Continuous**: Exposure duration is constantly adapted by the device to maximize the dynamic range.

On top of the previous standard values, a device might also provide device-specific values.

6 Digital I/O

Digital I/O covers the features required to control the general Input and Output signals of the camera. This includes Input control signals such as Triggers, Output signals such as Timer pulses but also static signals such as User configurable output bits.

The Digital I/O section models each I/O **Line** as a physical line that comes from the device connector and that goes into an **I/O Control Block** permitting to condition and to monitor the incoming or outgoing **Signal**.

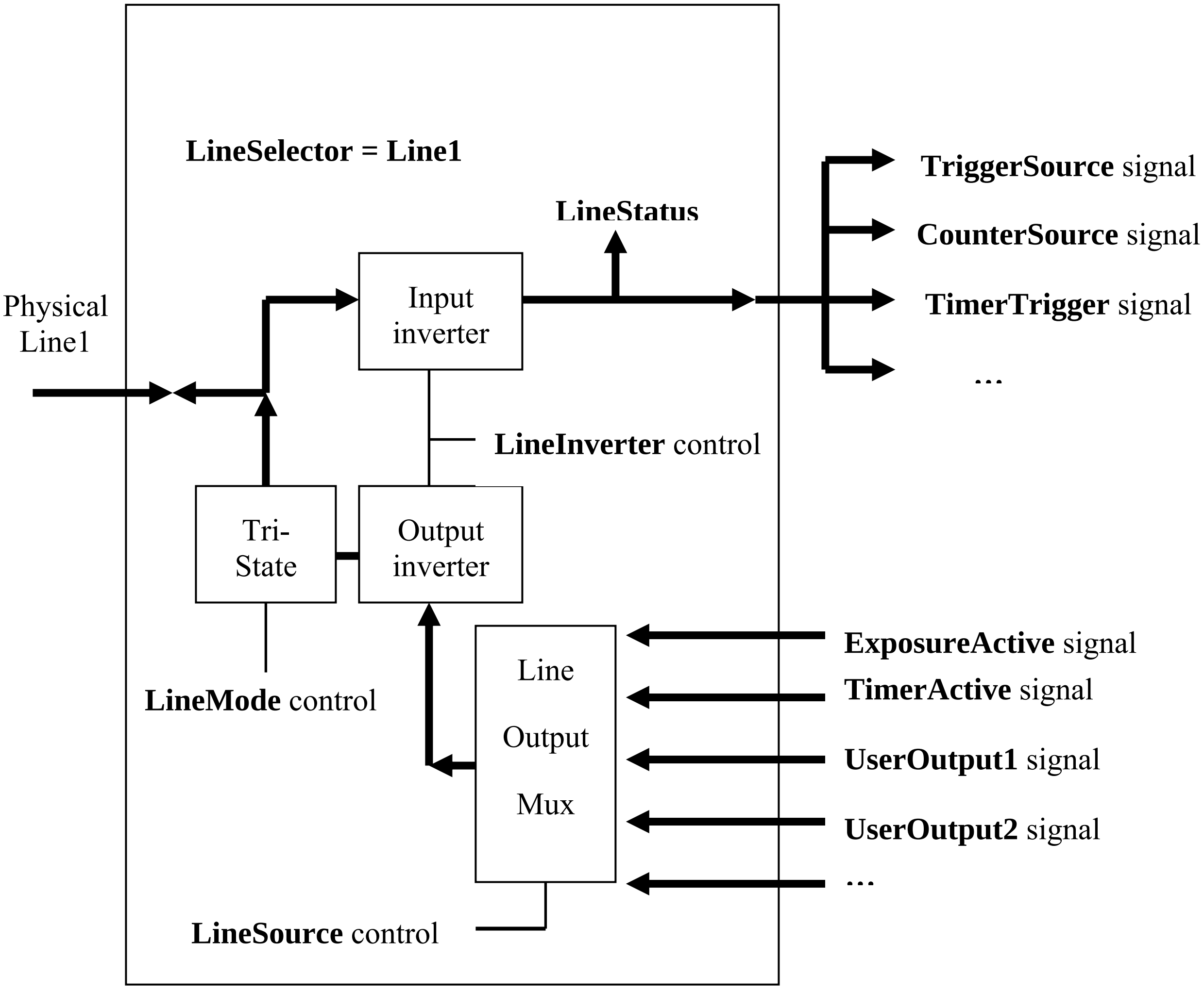

Figure 6-1: I/O Control

I/O Lines:

For a Digital I/O, when the full **I/O Control Block** is implemented, each physical **Line** (or pin) selected using **LineSelector** can be configured as Input or Output using **LineMode**. For an input or output Line, it is possible to read the Status of the Line with **LineStatus** and the incoming or outgoing signal can also be inverted using **LineInverter**. For an Output signal, the source of the signal is controlled using **LineSource** (See Figure 6-1).

For example, to output an inverted pulse coming from the Timer 1 on the physical Line 2 of the camera connector, use the following code:

```
LineSelector = Line2;

LineMode     = Output;

LineInverter = True;

LineSource   = Timer1Output;
```

Note that all the features of an I/O control block are optional. Typically, an Input only line will report the **LineMode** as **Input** (read-only) and will implement only the **LineSelector**, **LineInverter** and **LineStatus** features (top half in figure 6-1). An Output only line will report the **LineMode** as **Output** (read-only) and will implement only the **LineSelector**, **LineInverter** and **LineSource** features (bottom half of figure 6-1). Even a hard-wired input or output line is just particular case where all the features are read-only.

The electrical format of the physical Line (TTL. LVDS, Opto-Coupled…) can be read or controlled (if supported) using **LineFormat**.

Note also that the Status of all the Lines can be monitored in one single access using **LineStatusAll.**

UserOutput:

One possible source for Output lines is the User Output bit register.

Using **LineSource**, each of the bits of the User Output register can be directed to a physical output Line after going trough the I/O control block (See figure 6-1).

UserOutputSelector and **UserOutputValue** are used to set any individual bit of the User Output register. **UserOutputValueAll** and **UserOutputValueAllMask** can be used to set all or many of the User Output bits in one access.

6.1 LineSelector

Name	LineSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write

Unit	-
Recommended Visibility	Expert
Values	Line0 (If 0 based), Line1, Line2, …

This feature selects which physical line (or pin) of the external device connector to configure. When a Line is selected, all the other Line features will be applied to its associated I/O control block and will condition the resulting input or output signal.

LineSelector can take any of the following values:

- **Line0, Line1, Line2**: Index of the physical line and associated I/O control block to use.

6.2 LineMode

Name	LineMode[LineSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Input Output

This feature controls if the physical Line is used to Input or Output a signal. When a Line supports input and output mode, the default state is Input to avoid possible electrical contention.

LineMode can take any of the following values:

- **Input**: The selected physical line is used to Input an electrical signal.
- **Output**: The selected physical line is used to Output an electrical signal.

6.3 LineInverter

Name	LineInverter[LineSelector]
Level	Recommended
Interface	IBoolean
Access	Read/Write

Unit	-
Recommended Visibility	Expert
Values	False True

This feature controls if the electrical input or output signal on the selected Line is inverted.

LineInverter can take any of the following values:

- **False**: The Line signal is not inverted.
- **True**: The Line signal is inverted.

6.4 LineStatus

Name	LineStatus[LineSelector]
Level	Recommended
Interface	IBoolean
Access	Read
Unit	-
Recommended Visibility	Expert
Values	False True

This feature read the current status of the selected input or output Line. The status of the signal is taken after the input Line inverter of the I/O control block.

LineStatus can take any of the following values:

- **True**: The level of the Line signal is High.
- **False**: The level of the Line signal is Low.

6.5 LineStatusAll

Name	LineStatusAll
Level	Optional
Interface	IInteger
Access	Read

Unit	bitfield
Recommended Visibility	Expert
Values	Device-specific

Current logical state of all available Line signals at time of polling in a single bitfield. The order is Line0, Line1, Line2, …

6.6 LineSource

Name	LineSource[LineSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTriggerWait AcquisitionActive FrameTriggerWait FrameActive ExposureActive Timer1Active, Timer2Active, … Counter1Active, Counter2Active, … UserOutput0, UserOutput1, …

This feature is used to select which internal acquisition or I/O source signal to output on the selected Line when its **LineMode** is **Output.**

LineSource can take any of the following values (see Figure 6-1):

- **Off**: Line output is disabled (Tri-State).
- **AcquisitionTriggerWait**: Device is currently waiting for a trigger for the capture of one or many Frames.
- **AcquisitionActive**: Device is currently doing an acquisition of one or many Frames.
- **FrameTriggerWait**: Device is currently waiting for a Frame trigger.
- **FrameActive**: Device is currently doing the capture of a Frame.

- **ExposureActive**: Device is doing the exposure of a Frame (or Line).
- **Timer1Active, Timer2Active,** …: The chosen Timer is in active state.
- **Counter1Active, Counter2Active,** …: The chosen counter is in active state (counting).
- **UserOutput0, UserOutput1, UserOutput2,** …: The chosen User Output Bit state as defined by its current **UserOutputValue**.

6.7 LineFormat

Name	LineFormat[LineSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	NoConnect TriState TTL LVDS RS422 OptoCoupled

This feature returns or sets (if possible) the current electrical format of the selected physical input or output **Line**.

LineFormat can take any of the following values:

- **NoConnect**: The Line is not connected.
- **TriState:** The Line is currently in Tri-State mode (Not driven).
- **TTL:** The Line is currently accepting or sending TTL level signals.
- **LVDS:** The Line is currently accepting or sending LVDS level signals.
- **RS422:** The Line is currently accepting or sending RS422 level signals.
- **OptoCoupled**: The Line is Opto-Coupled.

6.8 UserOutputSelector

Name	UserOutputSelector
Level	Recommended

Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	UserOutput0 (If 0 based), UserOutput1, …

This feature selects which bit of the User Output register will be set by **UserOutputValue**.

UserOutputSelector can take any of the following values:

- **UserOutput0**: Selects the first bit of the User Output register (Bit 0).
- **UserOutput1**: Selects the first bit of the User Output register (Bit 1).
- **UserOutput2**: Selects the second bit of the User Output register (Bit 2).
- …

6.9 UserOutputValue

Name	UserOutputValue[UserOutputSelector]
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

This feature sets the value of the selected bit of the User Output register.

UserOutputValue can take any of the following values:

- **True**: Sets the bit to High.
- **False**: Sets the bit to Low.

6.10 UserOutputValueAll

Name	UserOutputValueAll

Level	Optional
Interface	IInteger
Access	Read/Write
Unit	bitfield
Recommended Visibility	Expert
Values	Device-specific

This feature sets the value of all the bits of the User Output register. It is subject to the **UserOutputValueAllMask**.

UserOutputValueAll can take any binary value and each bit set to one will set the corresponding User Output register bit to high. Note that the UserOutputs are numbered from 0 to N. This means that Bit 0 of **UserOutputValueAll** corresponds to the UserOutput0.

6.11 UserOutputValueAllMask

Name	UserOutputValueAllMask
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	bitfield
Recommended Visibility	Expert
Values	Device-specific

This feature sets the write mask to apply to the value specified by **UserOutputValueAll** before writing it in the User Output register. If the **UserOutputValueAllMask** feature is present, setting the user Output register using **UserOutputValueAll** will only change the bits that have a corresponding bit in the mask set to one.

UserOutputValueAllMask can take any binary value. Each bit set to one will enable writing of the corresponding User Output register bit and each bit set to zero will prevent it.

Note that **UserOutputValueAllMask** is ignored when an individual bit is set using **UserOutputValue**.

7 Counters and Timers Controls

This section lists all features that relates to control and monitoring of Counters and Timers.

A Counter is used to count internal events (FrameStart, FrameTrigger, …), I/O external events (Input Line rising edge, …) and even clock ticks. It can be Reset or Read at anytime. Counters and Timers can also be cascaded to increase their range if necessary.

Timers are readable and can be used to measure the duration of internal or external signals. A Timer can also be used to generate a timed strobe pulse with an optional delay before activation.

For example, to output a 300 us pulse coming from the Timer 1 when a rising edge trigger on the physical Line 2 of the camera connector happen, use the following code:

```
TimerSelector          = Timer1;
TimerDurationAbs       = 300;
TimerTriggerSource     = Line2;
TimerTriggerActivation = RisingEdge;
```

To set the destination output line of the Timer pulse, see the **LineSource** feature.

Note that Counters and Timers can also be used to generate an Event when a predetermined maximum count (or duration) is reached. See the **EventSelector** feature.

7.1 CounterSelector

Name	CounterSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Counter1, Counter2, …

This feature selects which counter to configure.

CounterSelector can take any of the following values:

- **Counter1**: Selects the first counter.
- **Counter2**: Selects the second counter.
- …

7.2 CounterEventSource

Name	CounterEventSource[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd LineStart LineEnd ExposureStart ExposureEnd Line0RisingEdge, Line1RisingEdge,… Counter1End, Counter2End, … Timer1End, Timer2End, … TimestampTick

This feature is used to select the events that will be the source to increment the counter.

CounterSource can take any of the following values (see Figure 5-1, Figure 5-2 and Figure 5-3):

- **Off**: Counter is stopped.
- **AcquisitionTrigger**: Counts the number of Acquisition Trigger.
- **AcquisitionStart**: Counts the number of Acquisition Start.
- **AcquisitionEnd**: Counts the number of Acquisition End.
- **FrameTrigger**: Counts the number of Frame Trigger.
- **FrameStart**: Counts the number of Frame start.
- **FrameEnd**: Counts the number of Frame end.
- **LineStart**: Counts the number of Line start.
- **LineEnd**: Counts the number of Line end.

- **ExposureStart**: Counts the number of Exposure start.
- **ExposureEnd**: Counts the number of Exposure end.
- **Line1RisingEdge, Line2RisingEdge,** …: Counts the number of rising edge transitions on the chosen I/O Line.
- **Counter1End, Counter2End,** …: Counts the number of Counter end when counter are cascaded.
- **Timer1End, Timer2End,** …: Counts the number of Timer pulses generated.
- **TimestampTick**: Counts the number of clock Ticks of the Timestamp clock. Can be used as a programmable timer.

7.3 CounterReset

Name	CounterReset[CounterSelector]
Level	Recommended
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Expert
Values	-

This feature is used to reset the selected counter. Note that the counter starts counting immediately after the reset. To disable the counter temporarily, set **CounterEventSource** to **Off**.

Note that the value of the Counter at time of reset is automatically latched and reflected in the **CounterValueAtReset.**

7.4 CounterValue

Name	CounterValue[CounterSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature is used to read the current value of the selected counter.

7.5 CounterValueAtReset

Name	CounterValueAtReset[CounterSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature is used to read the value of the selected counter when the counter was reset by a trigger or by an explicit **CounterReset** command. It represents the last counter value latched before to reset the counter.

7.6 CounterDuration

Name	CounterDuration[CounterSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature sets the duration (or number of events) before the **CounterEnd** event is generated. When the counter reaches the **CounterDuration** value, a **CounterEnd** event is generated, the **CounterActive** signal becomes inactive and the counter stops counting until a new trigger happens or it is explicitly reset with **CounterReset.**

7.7 CounterStatus

Name	CounterStatus[CounterSelector]

Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Expert
Values	CounterIdle CounterTriggerWait CounterActive CounterCompleted CounterOverflow

This feature is used to read the current state of the counter**.**

CounterStatus can take any of the following values:

- **CounterIdle**: The counter is idle. **CounterTriggerSource** is **Off**.
- **CounterTriggerWait**: The counter is waiting for a start trigger.
- **CounterActive**: The counter is counting for the specified duration.
- **CounterCompleted**: The counter reached the **CounterDuration** count.
- **CounterOverflow**: The counter reached its maximum possible count.

7.8 CounterTriggerSource

Name	CounterTriggerSource[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd ExposureStart

	ExposureEnd Line0, Line1, Line2, … Counter1End, Counter2End, … Timer1End, Timer2End, …

This feature is used to select the source to start the counter.

CounterTriggerSource can take any of the following values:

- **Off**: Disables the Timer trigger.
- **AcquisitionTrigger**: Starts with the reception of the Acquisition Trigger.
- **AcquisitionStart**: Starts with the reception of the Acquisition Start.
- **AcquisitionEnd**: Starts with the reception of the Acquisition End.
- **FrameTrigger**: Starts with the reception of the Frame Trigger.
- **FrameStart**: Starts with the reception of the Frame start.
- **FrameEnd**: Starts with the reception of the Frame end.
- **ExposureStart**: Starts with the reception of the Exposure start.
- **ExposureEnd**: Starts with the reception of the Exposure end.
- **Line0, Line1, Line2,** …: Starts with the reception of a transitions on the chosen I/O Line.
- **Counter1End, Counter2End,** …: Starts with the reception of the Counter end when counter are cascaded.
- **Timer1End, Timer2End,** …: Starts with the reception of the Timer end.

7.9 CounterTriggerActivation

Name	CounterTriggerActivation[CounterSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

This feature is used to select the type of activation for the trigger to start the counter.

CounterTriggerActivation can take any of the following values:

- **RisingEdge**: Starts counting on the Rising Edge of the selected trigger signal.
- **FallingEdge**: Starts counting on the Falling Edge of the selected trigger signal.
- **AnyEdge**: Starts counting on the Falling or rising Edge of the selected trigger signal.
- **LevelHigh**: Counts as long as the selected trigger signal level is High.
- **LevelLow**: Counts as long as the selected trigger signal level is Low.

7.10 TimerSelector

Name	TimerSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Timer1, Timer2, …

This feature selects which Timer to configure.

TimerSelector can take any of the following values:

- **Timer1:** Selects the first Timer.
- **Timer2:** Selects the second Timer.
- …

7.11 TimerDurationAbs

Name	TimerDurationAbs[TimerSelector]
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended	Expert

Visibility	
Values	≥0

This feature sets the duration (in microseconds) of the Timer pulse. When the Timer reaches the **TimerDurationAbs** value, a **TimerEnd** event is generated, the **TimerActive** signal becomes low and the Timer stops counting until a new trigger happens or it is explicitly reset with **TimerReset**.

TimerDurationRaw must reflect the state of **TimerDurationAbs** when they are both supported.

7.12 TimerDurationRaw

Name	TimerDurationRaw[TimerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature sets the duration in device-specific unit of the Timer pulse. When the Timer reaches the **TimerDurationRaw** value, a **TimerEnd** event is generated, the **TimerActive** signal becomes low and the Timer stops counting until a new trigger happens or it is explicitly reset with **TimerReset**.

TimerDurationAbs must reflect the state of **TimerDurationRaw** when they are both supported.

7.13 TimerDelayAbs

Name	TimerDelayAbs[TimerSelector]
Level	Recommended
Interface	IFloat
Access	Read/Write
Unit	us
Recommended Visibility	Expert
Values	≥0

This feature sets the duration (in microseconds) of the delay to apply after the reception of a trigger before starting the Timer pulse generation.

TimerDelayRaw must reflect the state of **TimerDelayAbs** when they are both supported.

7.14 TimerDelayRaw

Name	TimerDelayRaw[TimerSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature sets the duration in device-specific unit of the delay to apply after the reception of a trigger before starting the Timer pulse generation.

TimerDelayAbs must reflect the state of **TimerDelayRaw** when they are both supported.

7.15 TimerValueAbs

Name	TimerValueAbs[TimerSelector]
Level	Recommended
Interface	IFloat
Access	Read
Unit	us
Recommended Visibility	Expert
Values	≥0

This feature is used to read the current value (in microseconds) of the selected Timer.

7.16 TimerValueRaw

Name	TimerValueRaw[TimerSelector]
Level	Recommended

Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature is used to read the current value in device-specific unit of the selected Timer.

7.17 TimerStatus

Name	TimerStatus[TimerSelector]
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Expert
Values	TimerIdle TimerTriggerWait TimerActive TimerCompleted

This feature is used to read the current state of the Timer**.**

TimerStatus can take any of the following values:

- **TimerIdle**: The Timer is idle. **TimerTriggerSource** is **Off**.
- **TimerTriggerWait**: The Timer is waiting for a start trigger.
- **TimerActive**: The Timer is counting for the specified duration.
- **TimerCompleted**: The Timer reached the **TimerDuration** count.

7.18 TimerTriggerSource

Name	TimerTriggerSource[TimerSelector]
Level	Recommended
Interface	IEnumeration

Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd ExposureStart ExposureEnd Line0, Line1, … Counter1End, Counter2End, … Timer1End, Timer2End, …

This feature is used to select the source for the trigger to start the Timer.

TimerTriggerSource can take any of the following values:

- **Off**: Disables the Timer trigger.
- **AcquisitionTrigger**: Starts with the reception of the Acquisition Trigger.
- **AcquisitionStart**: Starts with the reception of the Acquisition Start.
- **AcquisitionEnd**: Starts with the reception of the Acquisition End.
- **FrameTrigger**: Starts with the reception of the Frame Trigger.
- **FrameStart**: Starts with the reception of the Frame start.
- **FrameEnd**: Starts with the reception of the Frame end.
- **ExposureStart**: Starts with the reception of the Exposure start.
- **ExposureEnd**: Starts with the reception of the Exposure end.
- **Line0, Line1, Line2,** …: Starts with the reception of a transition on the chosen I/O Line.
- **Counter1End, Counter2End,** …: Starts with the reception of the counter end.
- **Timer1End, Timer2End,** …: Starts with the reception of the Timer end when Timer are cascaded.

7.19 TimerTriggerActivation

Name	TimerTriggerActivation[TimerSelector]

Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

This feature is used to select the type of activation to start the Timer.

TimerTriggerActivation can take any of the following values:

- **RisingEdge**: Starts counting on the Rising Edge of the selected trigger signal.
- **FallingEdge**: Starts counting on the Falling Edge of the selected trigger signal.
- **AnyEdge**: Starts counting on the Falling or Rising Edge of the selected trigger signal.
- **LevelHigh**: Counts as long as the selected trigger signal level is High.
- **LevelLow**: Counts as long as the selected trigger signal level is Low.

8 Events Generation

This section describes how to control the generation of Events to the host application. An Event is a message that is sent to the host application to notify it of the occurrence of an internal event.

Events are typically used to synchronize the host application with some Events happening in the device. A typical use in machine vision is a host application that waits to be notified for the CCD exposure end to move the inspected part on a conveyer belt.

EventSelector select which particular Event to control or enable. There are 4 typical sources of event: Acquisition, Timer, Counter and I/O lines.

The standard Acquisition related Events are: **AcquisitionTrigger, AcquisitionStart, AcquisitionEnd, AcquisitionTransferStart, AcquisitionTransferEnd, AcquisitionError, FrameTrigger, FrameStart, FrameEnd, FrameTransferStart, FrameTransferEnd, ExposureStart, ExposureEnd** (see Figure 5-1, Figure 5-2 and Figure 5-3).

The standard Counters and Timers related Events are: **Counter1Start, Counter1End, Counter2Start, Counter2End, … Timer1Start, Timer End, Timer2Start, Timer2End,** …

The standard I/O line Events are: **Line0RisingEdge, Line0FallingEdge, Line0AnyEdge, Line1RisingEdge, Line1FallingEdge,** … Note that the event signal is monitored at the same place as **LineStatus** in the I/O control block (See Figure 6-1). This means that event is checked against the condition after the input inverter.

EventNotification is used to specify the type of notification to send (e.g: a standard GigEVision Event) when the internal event occurs. If **EventNotification** is **Off,** no event of the selected type is generated.

For example, to do a continuous acquisition and be notified at the end of the exposure period of each frame to move the part to be inspected, the following pseudo-code can be used:

```
Camera.AcquisitionMode   = Continuous;
Camera.EventSelector     = ExposureEnd;
Camera.EventNotification = GigEVisionEvent;
Camera.AcquisitionStart();
…
Camera.AcquisitionStop();
```

The list of recommended Event values is given in the table below.

8.1 EventSelector

Name	EventSelector
Level	Recommended

Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	AcquisitionTrigger AcquisitionStart AcquisitionEnd AcquisitionTransferStart AcquisitionTransferEnd AcquisitionError FrameTrigger FrameStart FrameEnd FrameTransferStart FrameTransferEnd ExposureStart ExposureEnd Counter1Start, … Counter1End, … Timer1Start, … Timer1End, … Line0RisingEdge, Line1RisingEdge, … Line0FallingEdge, Line1FallingEdge, … Line0AnyEdge, Line1AnyEdge, … Errors, … DeviceSpecificEvents …

This feature is used to select which internal Event to signal to the host application.

EventSelector can take any of the following values (see Figure 5-1, Figure 5-2, Figure 5-3 and Figure 6-1):

- **AcquisitionTrigger:** Device just received a trigger for the Acquisition of one or many Frames.
- **AcquisitionStart**: Device just started the Acquisition of one or many Frames.
- **AcquisitionEnd**: Device just completed the Acquisition of one or many Frames.
- **AcquisitionTransferStart**: Device just started the transfer of one or many Frames.

- **AcquisitionTransferEnd**: Device just completed the transfer of one or many Frames.
- **AcquisitionError**: Device just detected an error during the active Acquisition.
- **FrameTrigger**: Device just received a trigger for the capture of one Frame.
- **FrameStart**: Device just started the capture of one Frame.
- **FrameEnd**: Device just completed the capture of one Frame.
- **FrameTransferStart**: Device just started the transfer of one Frame.
- **FrameTransferEnd**: Device just completed the transfer of one Frame.
- **ExposureStart**: Device just started the exposure of one Frame (or Line).
- **ExposureEnd**: Device just completed the exposure of one Frame (or Line).
- **Counter1Start**: The event will be generated when counter 1 starts counting.
- **Counter1End**: The event will be generated when counter 1 ends counting.
- **Timer1Start**: The event will be generated when Timer 1 starts counting.
- **Timer1End**: The event will be generated when Timer 1 ends counting.
- **Line1RisingEdge**: The event will be generated when a Rising Edge is detected on the Line 1.
- **Line1FallingEdge**: The event will be generated when a Falling Edge is detected on the Line 1.
- **Line1AnyEdge**: The event will be generated when a Falling or Rising Edge is detected on the Line 1.
- …

8.2 EventNotification

Name	EventNotification[EventSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off GigEVisionEvent

This feature is used to select which type of notification is sent to the host application for the selected Event.

EventNotification can take any of the following values:

- **Off**: The selected Event notification is disabled.
- **GigEVisionEvent**: Sends a standard GigE Vision event notification.

9 Analog Controls

Features in this section describes how to influence the analog features of an image, such as gain, black level, white clip and gamma.

Some features provide 2 ways to set them: Raw and Absolute. Raw presents an integer value that is normally mapped into a register. Absolute presents a floating-point value that typically provides a more natural unit of measurement.

The **GainRaw/GainAbs**, **BlackLevelRaw/BalcklevelAbs** and **Gamma** features will transform the original pixel value Y to a new value Y' according to the following formula:

$$Y' = \left[\left(Y + BlackLevel\right) \cdot GainRaw\right]^{Gamma}$$

For some color cameras in Raw or RGB mode, the red/blue channel can be white balanced with respect to the green channel using the Red and blue **BalanceRatio** gain. For cameras in YUV mode the U/V channel can be balanced with respect to the Y channel using the U and V **BalanceRatio**, according to:

$$B' = B(BlueBalanceRatio \cdot GainRaw)$$

Other color camera controls each color channel gain independently, in which case, the Red, Green and Blue **GainRaw/GainAbs** features can be used for white balancing.

The automatic functions **GainAuto**, **BlackLevelAuto**, **BalanceWhiteAuto**, **GainAutoTapBalance** and **BlackLevelAutoTapBalance** can be used to auto-adjust a device once or continuously and to turn the function on and off.

Most of the automatic functions have 3 possible values: {**Off**, **Once**, **Continuous**}.

- **Off**: The automatic adjustement is disabled (ie. Manual control).
- **Once**: The automatic adjustement is performed once by the device. The affected features report the effective values. If necessary, the feature is automatically set to "Off" after the adjustment.
- **Continuous**: The automatic adjustement is continuously done by the device. The affected features report their effective values.

When a device has a specific auto-adjustment capability, it should have a corresponding feature allowing the necessary enumerations.

9.1 GainSelector

Name	GainSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	All Red Green Blue Y U V Tap1, Tap2, … AnalogAll AnalogRed AnalogGreen AnalogBlue AnalogY AnalogU AnalogV AnalogTap1, AnalogTap2, … DigitalAll DigitalRed DigitalGreen DigitalBlue DigitalY DigitalU DigitalV DigitalTap1, DigitalTap2, …

This feature selects which Gain is controlled by the various Gain features.

In general, there are 2 types of gain that can exit in a camera, analog or digital. Some camera will implement one or other or both. This is why there are 3 possible sets of gain.

The first one, without the **Analog** or **Digital** prefix, is to be used when only one type of gain is implemented. This permits to have an implementation independent way to set the gain.

The second and the third, with the **Analog** and **Digital** prefix, is to be used when both types of gain are implemented. This permits to have independent control over each one.

The possible values for **GainSelector** are:

- **All**: Gain will be applied to all channels or taps.
- **Red**: Gain will be applied to the red channel.
- **Green**: Gain will be applied to the green channel.
- **Blue**: Gain will be applied to the blue channel.
- **Y**: Gain will be applied to Y channel.
- **U**: Gain will be applied to U channel.
- **V**: Gain will be applied to V channel.
- **Tap1**: Gain will be applied to Tap 1.
- **Tap2**: Gain will be applied to Tap 2.
- …
- **AnalogAll**: Gain will be applied to all analog channels or taps.
- **AnalogRed**: Gain will be applied to the red analog channel.
- **AnalogGreen**: Gain will be applied to the green analog channel.
- **AnalogBlue**: Gain will be applied to the blue analog channel.
- **AnalogY**: Gain will be applied to Y analog channel.
- **AnalogU**: Gain will be applied to U analog channel.
- **AnalogV**: Gain will be applied to V analog channel.
- **AnalogTap1**: Analog gain will be applied to Tap 1.
- **AnalogTap2**: Analog gain will be applied to Tap 2.
- …
- **DigitalAll**: Gain will be applied to all digital channels or taps.
- **DigitalRed**: Gain will be applied to the red digital channel.
- **DigitalGreen**: Gain will be applied to the green digital channel.
- **DigitalBlue**: Gain will be applied to the blue digital channel.
- **DigitalY**: Gain will be applied to Y digital channel.
- **DigitalU**: Gain will be applied to U digital channel.

- **DigitalV**: Gain will be applied to V digital channel.
- **DigitalTap1**: Digital gain will be applied to Tap 1.
- **DigitalTap2**: Digital gain will be applied to Tap 2.
- …

9.2 GainRaw

Name	GainRaw[GainSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	Device-specific
Recommended Visibility	Beginner
Values	Device-specific

This feature controls the selected gain as a raw integer value. This is an amplification factor applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

GainAbs must reflect the value put in **GainRaw** when both features are supported.

For color or multi-tap cameras, **GainSelector** indicates the color channel or tap to control.

9.3 GainAbs

Name	GainAbs[GainSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	Device-specific
Recommended Visibility	Beginner
Values	Device-specific

This feature controls the selected gain as an absolute physical value. This is an amplification factor applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

GainRaw must reflect the value put in **GainAbs** when both features are supported.

For color or multi-tap cameras, **GainSelector** indicates the color channel or tap to control.

9.4 GainAuto

Name	GainAuto[GainSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Off Once Continuous Device-specific

This feature performs automatic gain control (AGC). The exact algorithm used to implement AGC is device-specific. Some other device-specific features might be used to allow the selection of the algorithm.

GainAuto can take any of the following values:

- **Off**: Gain is manually controlled using **GainRaw** or **GainAbs.**
- **Once**: Gain is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Gain is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.5 GainAutoBalance

Name	GainAutoBalance
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-

Recommended Visibility	Beginner
Values	Off Once Continuous Device-specific

This feature is used to perform automatic gain balancing between the sensor color channels or taps. The gain coefficients of each channel or tap are adjusted so they are matched.

GainAutoBalance can take any of the following values:

- **Off**: Gain tap balancing is manually controlled using **GainRaw** or **GainAbs**.
- **Once**: Gain tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Gain tap balancing is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.6 BlackLevelSelector

Name	BlackLevelSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	All Red Green Blue Y U V Tap1, Tap2, …

This feature selects which Black Level is controlled by the various Black Level features.

The possible values for **BlackLevelSelector** are:

- **All**: Black Level will be applied to all channels or taps.

- **Red**: Black Level will be applied to the red channel.
- **Green**: Black Level will be applied to the green channel.
- **Blue**: Black Level will be applied to the blue channel.
- **Y**: Black Level will be applied to Y channel.
- **U**: Black Level will be applied to U channel.
- **V**: Black Level will be applied to V channel.
- **Tap1**: Black Level will be applied to Tap 1.
- **Tap2**: Black Level will be applied to Tap 2.
- …

9.7 BlackLevelRaw

Name	BlackLevelRaw[BlackLevelSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	Device-specific
Recommended Visibility	Expert
Values	Device-specific

This feature controls the analog black level as a raw integer value. This represents a DC offset applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

BlackLevelAbs must reflect the value put in **BlackLevelRaw** when both features are supported.

For color or multi-tap cameras, **BlackLevelSelector** indicates which channel to access.

9.8 BlackLevelAbs

Name	BlackLevelAbs[BlackLevelSelector]
Level	Optional
Interface	IFloat
Access	Read/Write

Unit	Device-specific
Recommended Visibility	Expert
Values	Device-specific

This feature controls the analog black level as an absolute physical value. This represents a DC offset applied to the video signal.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

BlackLevelRaw must reflect the value put in **BlackLevelAbs** when both feature are supported.

For color or multi-tap cameras, **BlackLevelSelector** indicates which channel access.

9.9 BlackLevelAuto

Name	BlackLevelAuto[BlackLevelSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off Once Continuous Device-specific

This feature performs automatic black level adjustment. The exact algorithm used to implement this adjustment is device-specific. Some other device-specific features might be used to allow the selection of the algorithm.

BlackLevelAuto can take any of the following values:

- **Off**: Analog black level is manually controlled using **BlackLevelRaw** or **BlackLevelAbs**.
- **Once**: Analog black level is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Analog black level is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.10 BlackLevelAutoBalance

Name	BlackLevelAutoBalance
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off Once Continuous Device-specific

This feature is used to perform automatic black level balancing between the sensor color channels or taps. The black level coefficients of each channel are adjusted so they are matched.

BlackLevelAutoBalance can take any of the following values:

- **Off**: Black level tap balancing is manually controlled using **BlackLevelRaw** or **BlackLevelAbs**.
- **Once**: Black level tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: Black level tap balancing is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.11 WhiteClipSelector

Name	WhiteClipSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	All Red Green Blue

	Y U V Tap1, Tap2, …

This feature selects which White Clip is controlled by the various White Clip features.

The possible values for **WhiteClipSelector** are:

- **All**: White Clip will be applied to all channels or taps.
- **Red**: White Clip will be applied to the red channel.
- **Green**: White Clip will be applied to the green channel.
- **Blue**: White Clip will be applied to the blue channel.
- **Y**: White Clip will be applied to Y channel.
- **U**: White Clip will be applied to U channel.
- **V**: White Clip will be applied to V channel.
- **Tap1**: White Clip will be applied to Tap 1.
- **Tap2**: White Clip will be applied to Tap 2.
- …

9.12 WhiteClipRaw

Name	WhiteClipRaw[WhiteClipSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	Device-specific
Recommended Visibility	Expert
Values	Device-specific

This feature indicates the maximal intensity taken by the video signal before being clipped as a raw integer value. The video signal will never exceed the white clipping point: it will saturate at that level.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

WhiteClipAbs must reflect the value put in **WhiteClipRaw** when both feature are supported.

For color or multi-tap cameras, **WhiteClipTapSelector** indicates the channel to control.

9.13 WhiteClipAbs

Name	WhiteClipAbs[WhiteClipSelector]
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	Device-specific
Recommended Visibility	Expert
Values	Device-specific

This feature indicates the maximal intensity taken by the video signal before being clipped as an absolute physical value. The video signal will never exceed the white clipping point: it will saturate at that level.

The unit and values of this feature are specific to the device and must be defined in the XML device description file.

WhiteClipRaw must reflect the value put in **WhiteClipAbs** when both feature are supported.

For color or multi-tap cameras, **WhiteClipTapSelector** indicates the channel to control.

9.14 BalanceRatioSelector

Name	BalanceRatioSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Red Green Blue Y U V Tap1, Tap2, …

This feature selects which Balance ratio is controlled by the various Balance Ratio features.

The possible values for **BalanceRatioSelector** are:

- **Red**: Balance Ratio will be applied to the red channel.
- **Green**: Balance Ratio will be applied to the green channel.
- **Blue**: Balance Ratio will be applied to the blue channel.
- **Y**: Balance Ratio will be applied to Y channel.
- **U**: Balance Ratio will be applied to U channel.
- **V**: Balance Ratio will be applied to V channel.
- **Tap1**: Balance Ratio will be applied to Tap 1.
- **Tap2**: Balance Ratio will be applied to Tap 2.
- …

9.15 BalanceRatioAbs

Name	BalanceRatioAbs
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	>0.0

This feature is used for white balancing. It represents the ratio of the selected color component to a reference color component.

For example, the Color balance is realized by the following formula:

C_w = **BalanceRatioAbs** x C

where

C_w is the intensity of selected color component after white balancing.

BalanceRatioAbs is the white balance coefficient.

C is the intensity of the color component before white balancing.

9.16 BalanceWhiteAuto

Name	BalanceWhiteAuto
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Off Once Continuous Device-specific

This feature is used to perform automatic white balancing between the color channels. The white balancing ratios are automatically adjusted.

BalanceWhiteAuto can take any of the following values:

- **Off**: White balancing is manually controlled using **BalanceRatioSelector BalanceRatioAbs**.
- **Once**: White balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the **Off** state.
- **Continuous**: White balancing is constantly adjusted by the device.

On top of the previous standard values, a device might also provide device-specific values.

9.17 Gamma

Name	Gamma
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	>0.0

This feature is used to perform gamma correction of pixel intensity. This is typically used to compensate for non-linearity of the display system (such as CRT).

Gamma correction is realized by the following formula:

$$Y' = Y^{\mathbf{Gamma}}$$

where

Y' is the new pixel intensity

Y is the original pixel intensity

Gamma is the correction factor

The realization of the gamma correction can be implemented using a LUT. Therefore, it is possible that some LUT functionality is not available when gamma correction is activated.

10 LUT Controls

Features in this section describe the Look-up table (LUT) realated features.

10.1 LUTSelector

Name	LUTSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Luminance Red Green Blue Device-specific

This feature selects which LUT is controlled by the various LUT features. It is typically not available when only a single LUT is supported.

The selector must be changed prior to accessing the features it indexes.

10.2 LUTEnable

Name	LUTEnable[LUTSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

This feature activates the selected LUT.

For cameras supporting multiple LUT, **LUTSelector** indicates the LUT to access.

10.3 LUTIndex

Name	LUTIndex[LUTSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

This feature provides the index (offset) of the coefficient to access in the selected LUT.

For cameras supporting multiple LUT, **LUTSelector** indicates the LUT to access.

10.4 LUTValue

Name	LUTValue[LUTSelector][LUTIndex]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	Device-specific

This feature represents the Value found at entry **LUTIndex** of the LUT selected by **LUTSelector**.

10.5 LUTValueAll

Name	LUTValueAll[LUTSelector]
Level	Optional
Interface	IRegister
Access	Read/Write
Unit	-
Recommended	Guru

Visibility	
Values	Device-specific

This feature allows streaming all the LUT coefficients without having to use the **LUTIndex** feature. For cameras supporting multiple LUT, **LUTSelector** indicates the LUT to access.

11 GigE Vision Transport Layer

This use case provides access to GigE Vision bootstrap registers and other information related to the GigE Vision transport medium. Note most of these registers are mapped according to GigE Vision specification.

In most situations, these registers are directly handled by the framework managing the transport layer on the PC and are not directly visible to user (for example, deciding which UDP port number to use for a stream channel).

More information about exact meaning of these features is found in the GigE Vision specification. The GigE Vision specification shall have precedence over this list in case of disparity.

Convention for this section:

- All GigE Vision features start with the “Gev” prefix
- GigE Vision registers are 32-bit. If a GigE Vision register has multiple fields within this 32-bit, then they are separated in multiple features.
- If the user has configured the camera front end he can read from the back end that **PayloadSize** will be transferred for each image. This number covers all kind of data coming with the image, e.g. stamps etc. If the user allocates **PayloadSize** for each buffer he can be sure that each frame will fit into his buffers.

11.1 PayloadSize

Name	PayloadSize
Level	Mandatory
Interface	IInteger
Access	Read-only
Unit	bytes
Recommended Visibility	Expert
Values	>0

PayloadSize provides the number of bytes transferred for each image on the stream channel, including any end-of-line, end-of-frame statistics or other stamp data. This is the total size of data payload for a block. UDP and GVSP headers are not considered. Data leader and data trailer are not included.

This is mainly used by the application software to determine size of image buffers to allocate (largest buffer possible for current mode of operation).

For example, an image with no statistics or stamp data as **PayloadSize** equals to (width x height x pixel size) in bytes. It is strongly recommended to retrieve **PayloadSize** from the camera instead of relying on the above formula.

11.2 GevVersionMajor

Name	GevVersionMajor
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	>0

This field represents the major version of the specification. For instance, GigE Vision version 1.0 would have the major version set to 1.

11.3 GevVersionMinor

Name	GevVersionMinor
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	≥0

This field represents the minor version of the specification. For instance, GigE Vision version 1.0 would have the minor version set to 0.

11.4 GevDeviceModelIsBigEndian

Name	GevDeviceModelIsBigEndian

Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True False

Endianess might be used to interpret multi-byte data for READMEM and WRITEMEM commands. This represents the endianess of bootstrap registers.

Note this bit has no effect on the endianess of the GigE Vision protocol headers: they are always big-endian.

11.5 GevDeviceModeCharacterSet

Name	GevDeviceModeCharacterSet
Level	Optional
Interface	IEnumeration
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	UTF8

This feature represents the character set used by all the strings of the bootstrap registers.

11.6 GevInterfaceSelector

Name	GevInterfaceSelector
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert

Values	≥0

This feature is a Selector that indicates to which physical network interface other features must reference.

11.7 GevMACAddress

Name	GevMACAddress[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Beginner
Values	≥0

This feature stores the MAC address of the given network interface. This feature must return a 64-bit value representing the full MAC address of the device i.e. the high and low parts.

11.8 GevSupportedIPConfigurationLLA

Name	GevSupportedIPConfigurationLLA[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	True False

This feature indicates if Link Local Address IP configuration scheme is supported by the given network interface.

11.9 GevSupportedIPConfigurationDHCP

Name	GevSupportedIPConfigurationDHCP[GevInterfaceSelector]

Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	True False

This feature indicates if DHCP IP configuration scheme is supported by the given network interface.

11.10GevSupportedIPConfigurationPersistentIP

Name	GevSupportedIPConfigurationPersistentIP[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	True False

This feature indicates if PersistentIP configuration scheme is supported by the given network interface.

11.11GevCurrentIPConfiguration

Name	GevCurrentIPConfiguration[GevInterfaceSelector]
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended	Beginner

Visibility	
Values	PersistentIP DHCP LLA

This feature reports the current IP Configuration scheme. Note that this feature doesn’t provision more that one simultaneous IP configuration and should not be used. This feature is part of this document only for backward compatibility reasons. This feature is deprecated. It has been replaced by GevCurrentIPConfigurationLLA, GevCurrentIPConfigurationDHCP and GevCurrentIPConfigurationPersistentIP.

11.12GevCurrentIPConfigurationLLA

Name	GevCurrentIPConfigurationLLA[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	True

This feature indicates if Link Local Address IP configuration scheme is activated on the given network interface.

11.13GevCurrentIPConfigurationDHCP

Name	GevCurrentIPConfigurationDHCP[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	True False

This feature indicates if DHCP IP configuration scheme is activated on the given network interface.

11.14GevCurrentIPConfigurationPersistentIP

Name	GevCurrentIPConfigurationPersistentIP[GevInterfaceSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	True False

This feature indicates if PersistentIP configuration scheme is activated on the given network interface.

11.15GevCurrentIPAddress

Name	GevCurrentIPAddress[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Beginner
Values	≥0

This feature reports the IP address for the given network interface once it has been configured.

11.16GevCurrentSubnetMask

Name	GevCurrentSubnetMask[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read-only

Unit	-
Recommended Visibility	Beginner
Values	≥0

This feature provides the subnet mask of the given interface.

11.17GevCurrentDefaultGateway

Name	GevCurrentDefaultGateway[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Beginner
Values	≥0

This feature indicates the default gateway IP address to be used on the given network interface.

11.18GevFirstURL

Name	GevFirstURL
Level	Optional
Interface	IString
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	-

This feature stores the first URL to the XML device description file. The First URL is used as the first choice by the application to retrieve the XML device description file.

11.19GevSecondURL

Name	GevSecondURL
Level	Optional
Interface	IString
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	-

This feature stores the second URL to the XML device description file. This URL is an alternative if the application was unsuccessful to retrieve the device description file using the first URL.

11.20GevNumberOfInterfaces

Name	GevNumberOfInterfaces
Level	Optional
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	>0

This feature indicates the number of physical network interfaces supported by this device.

11.21GevPersistentIPAddress

Name	GevPersistentIPAddress[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended	Expert

Visibility	
Values	≥0

This feature indicates the Persistent IP address for this network interface. It is only used when the device boots with the Persistent IP configuration scheme.

11.22GevPersistentSubnetMask

Name	GevPersistentSubnetMask[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature indicates the Persistent subnet mask associated with the Persistent IP address on this network interface. It is only used when the device boots with the Persistent IP configuration scheme.

11.23GevPersistentDefaultGateway

Name	GevPersistentDefaultGateway[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature indicates the persistent default gateway for this network interface. It is only used when the device boots with the Persistent IP configuration scheme.

11.24GevMessageChannelCount

Name	GevMessageChannelCount
Level	Optional
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	0 or 1

This feature reports the number of message channels supported by this device.

11.25GevStreamChannelCount

Name	GevStreamChannelCount
Level	Optional
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	1 to 512

This feature reports the number of stream channels supported by this device.

11.26GevSupportedOptionalCommandsUserDefinedName

Name	GevSupportedOptionalCommandsUserDefinedName
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True

	False

User-defined name register is supported.

11.27GevSupportedOptionalCommandsSerialNumber

Name	GevSupportedOptionalCommandsSerialNumber
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True False

Serial number register is supported.

11.28GevSupportedOptionalCommandsEVENTDATA

Name	GevSupportedOptionalCommandsEVENTDATA
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True False

EVENTDATA_CMD and EVENTDATA_ACK are supported.

11.29GevSupportedOptionalCommandsEVENT

Name	GevSupportedOptionalCommandsEVENT
Level	Optional

Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True False

EVENT_CMD and EVENT_ACK are supported.

11.30 GevSupportedOptionalCommandsPACKETRESEND

Name	GevSupportedOptionalCommandsPACKETRESEND
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True False

PACKETRESEND_CMD is supported.

11.31 GevSupportedOptionalCommandsWRITEMEM

Name	GevSupportedOptionalCommandsWRITEMEM
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True False

WRITEMEM_CMD and WRITEMEM_ACK are supported.

11.32GevSupportedOptionalCommandsConcatenation

Name	GevSupportedOptionalCommandsConcatenation
Level	Optional
Interface	IBoolean
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	True False

Multiple operations in a single message are supported.

11.33GevHeartbeatTimeout

Name	GevHeartbeatTimeout
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	ms
Recommended Visibility	Guru
Values	>0

This feature indicates the current heartbeat timeout in milliseconds.

11.34GevTimestampTickFrequency

Name	GevTimestampTickFrequency
Level	Optional
Interface	IInteger
Access	Read-only
Unit	ticks

Recommended Visibility	Expert
Values	≥0

This 64-bit feature indicates the number of timestamp ticks during 1 second. This corresponds to the timestamp frequency in Hertz.

11.35GevTimestampControlLatch

Name	GevTimestampControlLatch
Level	Optional
Interface	ICommand
Access	Write-only
Unit	-
Recommended Visibility	Expert
Values	-

Latch current timestamp counter into “Timestamp value” register.

11.36GevTimestampControlReset

Name	GevTimestampControlReset
Level	Optional
Interface	ICommand
Access	Write-only
Unit	-
Recommended Visibility	Expert
Values	-

Reset timestamp 64-bit counter to 0.

11.37GevTimestampValue

Name	GevTimestampValue

Level	Optional
Interface	IInteger
Access	Read
Unit	ticks
Recommended Visibility	Expert
Values	≥0

This feature reports the latched 64-bit value of the timestamp counter. It is necessary to latch the 64-bit timestamp value to guaranty its integrity when performing the two 32-bit read accesses to retrieve the higher and lower 32-bit portions.

11.38GevCCP

Name	GevCCP
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	OpenAccess ExclusiveAccess ControlAccess

This feature is used to grant privilege to an application. Only one application is allowed to control the device. This application is able to write into device's registers. Other applications can read device's register only if the controlling application does not have the exclusive privilege.

11.39GevMCPHostPort

Name	GevMCPHostPort
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-

Recommended Visibility	Guru
Values	≥0

Indicates the port to which the device must send messages. Setting this value to 0 closes the message channel.

11.40GevMCDA

Name	GevMCDA
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

This feature indicates the destination IP address for the message channel.

11.41GevMCTT

Name	GevMCTT
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	ms
Recommended Visibility	Guru
Values	>0

This feature provides the transmission timeout value in milliseconds.

11.42GevMCRC

Name	GevMCRC

Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

This feature indicates the number of retransmissions allowed when a message channel message times out.

11.43GevStreamChannelSelector

Name	GevStreamChannelSelector
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

This Selector is used to index into the various stream channel features.

11.44GevSCPInterfaceIndex

Name	GevSCPInterfaceIndex[GevStreamChannelSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Index of network interface to use (from 0 to 3). Specific streams might be hard-coded to a specific network interfaces. Therefore this field might not be programmable on certain devices. It is read-only for this case.

11.45GevSCPHostPort

Name	GevSCPHostPort[GevStreamChannelSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

Indicates the port to which the device must send data stream. Setting this value to 0 closes the stream channel.

11.46GevSCPSFireTestPacket

Name	GevSCPSFireTestPacket[GevStreamChannelSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

When this bit is set, the device will fire one test packet. The “don’t fragment” bit of IP header must be set for this test packet.

11.47GevSCPSDoNotFragment

Name	GevSCPSDoNotFragment[GevStreamChannelSelector]
Level	Optional

Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

This bit is copied into the “do not fragment” bit of IP header of each stream packet. It can be used by the application to prevent IP fragmentation of packets on the stream channel.

11.48GevSCPSBigEndian

Name	GevSCPSBigEndian[GevStreamChannelSelector]
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	True False

Endianess of multi-byte pixel data for this stream.

This is an optional feature. A device that does not support this feature must support little-endian and always leave that bit clear.

11.49GevSCPSPacketSize

Name	GevSCPSPacketSize[GevStreamChannelSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	bytes
Recommended Visibility	Expert

Values	>0

The stream packet size to send on this channel, except for data leader and data trailer; and the last data packet which might be of smaller size (since packet size is not necessarily a multiple of block size for stream channel). The value is in bytes.

If a device cannot support the requested packet size, then it must not fire a test packet when requested to do so.

11.50GevSCPD

Name	GevSCPD[GevStreamChannelSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	ticks
Recommended Visibility	Expert
Values	≥0

This feature indicates the delay (in timestamp counter unit) to insert between each packet for this stream channel. This can be used as a crude flow-control mechanism if the application cannot keep up with the packets coming from the device.

11.51GevSCDA

Name	GevSCDA[GevStreamChannelSelector]
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	≥0

This feature indicates the destination IP address for this stream channel.

11.52GevLinkSpeed

Name	GevLinkSpeed[GevInterfaceSelector]
Level	Optional
Interface	IInteger
Access	Read-only
Unit	Mbps
Recommended Visibility	Expert
Values	>0

This feature indicates the speed of transmission negotiated by the given network interface.

11.53GevIPConfigurationStatus

Name	GevIPConfigurationStatus[GevInterfaceSelector]
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Beginner
Values	None PersistentIP DHCP LLA ForceIP

This feature reports the current IP configuration status.

12 User Sets

This section describes the features for global control of the device settings. It allows loading or saving factory or user-defined settings.

Loading the factory default User Set guarantees a state where a continuous acquisition can be started using only the mandatory features.

12.1 UserSetSelector

Name	UserSetSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Default UserSet1, UserSet2, …

Selects the feature User Set to load, save or configure.

Possible values for **UserSetSelector** are:

- **Default**: Selects the factory setting User set.
- **UserSet1**: Selects the first user set.
- **UserSet2**: Selects the second user set.
- …

When **Default** User Set is selected and loaded using **UserSetLoad**, the device must be in default factory settings state and must make sure the mandatory continuous acquisition use case works directly. Default User Set is read-only and cannot be modified.

12.2 UserSetLoad

Name	UserSetLoad[UserSetSelector]
Level	Recommended
Interface	ICommand
Access	Write-only
Unit	-

Recommended Visibility	Beginner
Values	-

Loads the User Set specified by **UserSetSelector** to the device and makes it active.

12.3 UserSetSave

Name	UserSetSave[UserSetSelector]
Level	Recommended
Interface	ICommand
Access	Write-only
Unit	-
Recommended Visibility	Beginner
Values	-

Save the User Set specified by **UserSetSelector** to the non-volatile memory of the device.

12.4 UserSetDefaultSelector

Name	UserSetDefaultSelector
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Beginner
Values	Default UserSet1, UserSet2, …

Selects the feature User Set to load and make active when the device is reset.

Possible values for **UserSetDefaultSelector** are:

- **Default**: Select the factory setting User set.

- **UserSet1**: Select the first User Set.
- **UserSet2**: Select the second User Set.
- …

If **Default** is selected, the device will boot with the default factory settings and makes sure the mandatory continuous acquisition use case works directly.

13 Chunk Data Streams

Chunks are tagged blocks of data. The tags allow a chunk parser to dissect the data payload into its elements and to identify the content.

The length of a frame varies depending on the number of activated chunks, but the user can always expect a frame with the maximum size of **PayloadSize**.

With chunks disabled by setting **ChunkModeActive** to **False** the camera streams frames consisting only of the image.

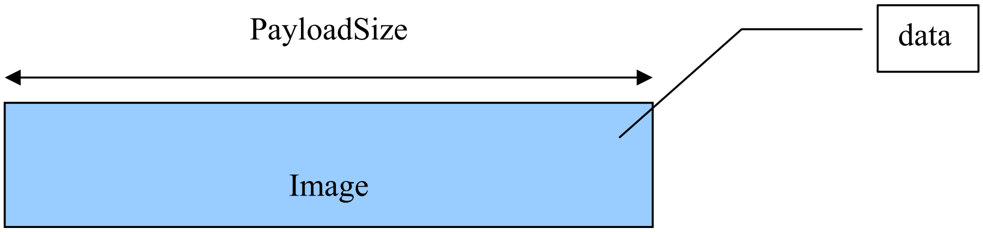

Figure 13-1: Frame with chunks disabled

With chunks enabled by setting **ChunkModeActive** to **True** the camera streams frames consisting of chunks. In this mode the image is a chunk too.

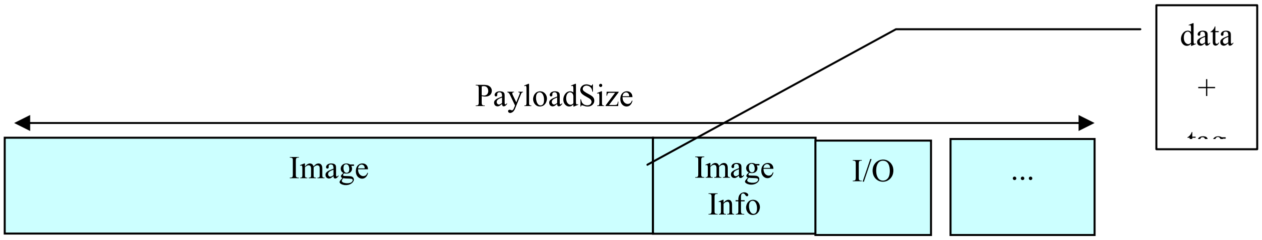

Figure 13-2: Frame with chunks enabled

Each chunk can be enabled or disabled using the **ChunkSelector** and **ChunkEnable** feature. This allows controlling the embedding of different information in the payload.

For example, a possible value for a chunk is **ImageInformation** that embeds all the information describing the current Image. e.g.: **Width**, **Height**, **OffsetX**, **OffsetY**, **PixelFormat**, **PixelDynamicRangeMin**, **PixelDynamicRangeMax**, …

The data in the chunks is exposed via the chunk parser. The naming scheme to access the data of the chunk *name* is **Chunk***name*.

13.1 ChunkModeActive

Name	ChunkModeActive
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

This feature activates the inclusion of Chunk data in the payload of the image.

13.2 ChunkSelector

Name	ChunkSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Image OffsetX OffsetY Width Height PixelFormat DynamicRangeMax DynamicRangeMin Timestamp LineStatusAll

This feature selects which Chunk to enable or control.

13.3 ChunkEnable

Name	ChunkEnable[ChunkSelector]

Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	True False

This feature enables the inclusion of the selected Chunk data in the payload of the image.

13.4 ChunkImage

Name	ChunkImage
Level	Recommended
Interface	IRegister
Access	Read-only
Unit	-
Recommended Visibility	Guru
Values	Device-specific

This feature returns the entire image data included in the payload.

13.5 ChunkOffsetX

Name	ChunkOffsetX
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	pixels
Recommended Visibility	Expert
Values	≥0

This feature returns the **OffsetX** of the image included in the payload.

13.6 ChunkOffsetY

Name	ChunkOffsetY
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	pixels
Recommended Visibility	Expert
Values	≥0

This feature returns the **OffsetY** of the image included in the payload.

13.7 ChunkWidth

Name	ChunkWidth
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	Pixels
Recommended Visibility	Expert
Values	>0

This feature returns the **Width** of the image included in the payload.

13.8 ChunkHeight

Name	ChunkHeight
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	Pixels
Recommended	Expert

Visibility	
Values	>0

This feature returns the **Height** of the image included in the payload.

13.9 ChunkPixelFormat

Name	ChunkPixelFormat
Level	Recommended
Interface	IEnumeration
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	Mono8 Mono8Signed Mono10 Mono10Packed Mono12 Mono12Packed Mono16 BayerGR8 BayerRG8 BayerGB8 BayerBG8 BayerGR10 BayerRG10 BayerGB10 BayerBG10 BayerGR12 BayerRG12 BayerGB12 BayerBG12 RGB8Packed BGR8Packed RGBA8Packed BGRA8Packed RGB10Packed BGR10Packed RGB12Packed BGR12Packed

	RGB10V1Packed RGB10V2Packed YUV411Packed YUV422Packed YUV444Packed RGB8Planar RGB10Planar RGB12Planar RGB16Planar Device-specific

This feature returns the **PixelFormat** of the image included in the payload.

13.10ChunkDynamicRangeMin

Name	ChunkDynamicRangeMin
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature returns the minimum value of dynamic range of the image included in the payload.

13.11ChunkDynamicRangeMax

Name	ChunkDynamicRangeMax
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature returns the maximum value of dynamic range of the image included in the payload.

13.12ChunkTimestamp

Name	ChunkTimestamp
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature returns the Time stamp of the image included in the payload at the time of the FrameStart internal event (see Figure 5-2).

13.13ChunkLineStatusAll

Name	ChunkLineStatusAll
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	bitfield
Recommended Visibility	Expert
Values	≥0

This feature returns the status of all the I/O lines at the time of the FrameStart internal event (see Figure 5-2).

13.14ChunkCounterSelector

Name	ChunkCounterSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write

Unit	-
Recommended Visibility	Expert
Values	Counter1, Counter2,…

This feature selects the Counter to read with ChunkCounter.

The standard values are: **Counter1, Counter2**,…

13.15ChunkCounter

Name	ChunkCounter[ChunkCounterSelector]
Level	Recommended
Interface	IInteger
Access	Read-only
Unit	-
Recommended Visibility	Expert
Values	≥0

This feature returns the value of the selected Chunk counter at the time of the FrameStart internal event (see Figure 5-2).

13.16ChunkTimerSelector

Name	ChunkTimerSelector
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Expert
Values	Timer1, Timer2, …

This feature selects the Timer to read with ChunkTimer.

The standard value are: **Timer1, Timer2**,…

13.17ChunkTimer

Name	ChunkTimer[ChunkTimerSelector]
Level	Recommended
Interface	IFloat
Access	Read-only
Unit	us
Recommended Visibility	Expert
Values	>0

This feature returns the value of the selected Timer at the time of the FrameStart internal event (See Figure 5-2).

14 File Access Controls

The File Access Controls section describes all features related to accessing files in the device.

It contains the definition of a generic file access schema for GenICam compliant devices. It is based on a set of standard features that are controlled from adapter code which resides in the GenICam reference implementation. The adapter code presents its services through an interface inherited from std::iostream.

The model, on which the controls are based, is depicted in the following diagram

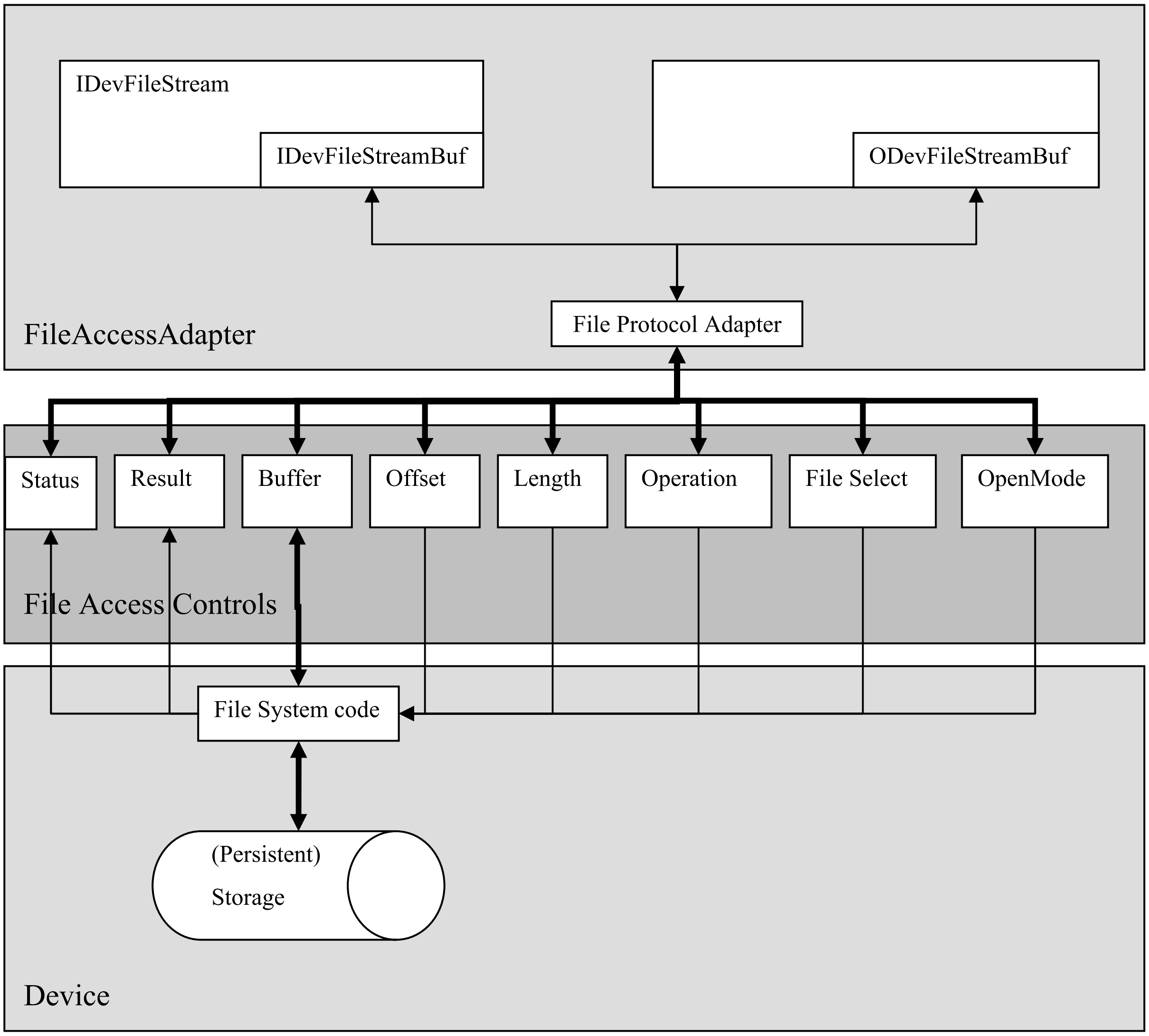

Figure 14-1: File Access Model

It assumes that all operations, which can be done on the persistent storage, could be executed by using operations with the semantic of fopen/fclose/fread/fwrite. The operations and their parameters are mapped onto the features of the list of File Access Controls.

To provide a generic API on top of the File Access Controls, a FileAccessAdapter is defined in the GenApi. The Adapter provides two iostream interfaces to the device files:

- **IDevFileStream** Read from the device
- **ODevFileStream** Write to the device

The File Protocol Adapter is responsible for the mapping of the (I/O) DevFileStreamBuf actions Open, Close, UnderFlow, Overflow on File Access Controls

Example Code for the streaminterface:

```
//GenApi::INodeMap * pInterface
ODevFileStream usersetWrite;
usersetWrite.open(pInterface, "UserSet1");
if( ! usersetWrite.fail() ){
     usersetWrite << "Hello World\n";
}
usersetWrite.close();

IDevFileStream usersetRead;
usersetRead.open(pInterface, "UserSet1");
if( ! usersetRead.fail() ){
     cout << usersetRead.rdbuf();
}
usersetRead.close();
```

File Access Control features:

The **FileSelector** feature selects the target file in the device for the Operation. The entries of this enumeration define the names of all files in the device that can be accessed via the File Access.

FileOperationSelector specifies the operation to execute on the file.

FileOperationExecute command starts the selected operation execution.

FileOpenMode is a parameter for the Open operation and controls the access mode (Read, Write, ReadWrite) in which the file is opened.

FileOperationStatus returns the status of the last operation executed on the file. This feature must return Success if the operation is executed as requested.

FileOperationResult returns the number of bytes successfully read/written bytes during the previous Read or Write operations.

FileSize returns the size of the file in bytes.

The data, that is read from or written to the device, is exchanged between the application and the device through the **FileAccessBuffer** feature. This register mapped **FileAccessBuffer** must be written with the target data before to execute the Write operation using **FileOperationExecute**. For Read operation, the data can be read from the **FileAccessBuffer** after the Read operation has been executed.

FileAccessOffset controls the starting position of the access in the file**.**

FileAccessLength controls the number of bytes to transfer to or from the **FileAccessBuffer** during the following Read or Write operation.

Altogether, the features **FileSelector**, **FileAccessOffset** and **FileAccessLength** control the mapping between the device file storage and the **FileAccessBuffer**.

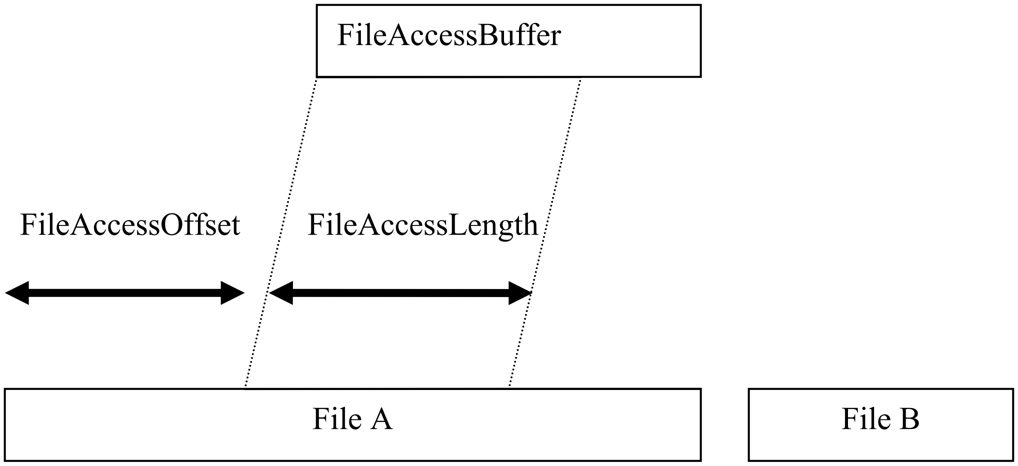

Figure 14-2: Layout of FileAccessBuffer (FileA is selected by FileSelector)

14.1 FileSelector

Name	FileSelector
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Recommended Visibility	Guru
Values	UserSetDefault UserSet1 UserSet2 UserSet3 … LUTLuminance LUTRed LUTGreen LUTBlue …

The **FileSelector** feature 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.

FileSelector can take any of the following values:

- **UserSetDefault**: the default user set of the device
- **UserSet1**: the first user set of the device
- **UserSet2**: the second user set of the device
- **UserSet3**: the third user set of the device
- …
- **LUTLuminance**: The Luminance LUT of the camera.
- **LUTRed**: The Red LUT of the camera.
- **LUTGreen**: The Green LUT of the camera.
- **LUTBlue**: The Blue LUT of the camera.
- …

On top of the previous standard values, a device might also provide device-specific values.

14.2 FileOperationSelector

Name	FileOperationSelector[FileSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Recommended Visibility	Guru
Values	Open Close Read Write

The **FileOperationSelector** feature selects the target operation for the selected file in the device. This Operation is executed when the **FileOperationExecute** feature is called.

FileOperationSelector can take any of the following values:

- **Open**: Opens the file selected by **FileSelector** in the device. The access mode in which the file is opened is selected by **FileOpenMode**

- **Close**: Closes the file selected by **FileSelector** in the device.
- **Read**: Reads **FileAccessLength** bytes from the device storage at the file relative offset **FileAccessOffset** into **FileAccessBuffer.**
- **Write:** Writes **FileAccessLength** bytes taken from the **FileAccessBuffer** into the device storage at the file relative offset **FileAccessOffset.**

14.3 FileOperationExecute

Name	FileOperationExecute[FileSelector][FileOperationSelector]
Level	Recommended
Interface	ICommand
Access	Write
Unit	-
Recommended Visibility	Guru
Values	-

The **FileOperationExecute** feature is the command that executes the operation selected by **FileOperationSelector** on the selected file.

14.4 FileOpenMode

Name	FileOpenMode[FileSelector]
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Guru
Values	Read Write ReadWrite

The **FileOpenMode** feature selects the access mode in which a file is opened in the device.

FileOpenMode can take any of the following values:

- **Read:** This mode selects read-only open mode.

- **Write:** This mode selects write-only open mode.
- **ReadWrite:** This mode selects read and write open mode.

14.5 FileAccessBuffer

Name	FileAccessBuffer
Level	Recommended
Interface	IRegister
Access	Read/(Write)
Unit	-
Recommended Visibility	Guru
Values	Device-specific

The **FileAccessBuffer** feature defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.

This register mapped **FileAccessBuffer** must be written with the target data before to execute a Write operation. For Read Operation, the data can be read from the **FileAccessBuffer** after the Read operation has been executed. The effective data transfer is done upon **FileOperationExecute** execution (See figure 14-2).

14.6 FileAccessOffset

Name	FileAccessOffset[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	Byte
Recommended Visibility	Guru
Values	>= 0

This feature controls the mapping between the device file storage and the **FileAccessBuffer**.

The **FileAccessOffset** defines the offset in bytes of the **FileAccessBuffer** relative to the beginning of the selected File (See figure 14-2). This feature is available only when **FileOperationSelector** is set to Read or Write.

14.7 FileAccessLength

Name	FileAccessLength[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	Byte
Recommended Visibility	Guru
Values	>= 0

This feature controls the mapping between the device file storage and the **FileAccessBuffer.**

The **FileAccessLength** defines the number of bytes to transfer to or from the **FileAccessBuffer** (See figure14-2). This feature is available only when **FileOperationSelector** is set to Read or Write.

14.8 FileOperationStatus

Name	FileOperationStatus[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Recommended Visibility	Guru
Values	Success (mandatory) Failure …

The **FileOperationStatus** feature represents the file operation execution status. Upon execution of a successful file operation, it must return **Success**. In case of complete or partial failure of the operation, other return values can be defined to indicate the nature of the error that happened. If only one fail status is defined, it should be defined as **Failure**.

14.9 FileOperationResult

Name	FileOperationResult[FileSelector][FileOperationSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Recommended Visibility	Guru
Values	-

The **FileOperationResult** feature represents the file operation result. For Read or Write operations, the number of successfully read/written bytes is returned.

14.10 FileSize

Name	FileSize[FileSelector]
Level	Recommended
Interface	IInteger
Access	Read
Unit	Byte
Recommended Visibility	Guru
Values	>=0

The **FileSize** feature represents the size of the selected file in bytes.

15 Typical Standard Feature usage examples

This section shows examples of typical use cases of the standard acquisition features in C/C++ pseudo-code.

For simplicity, the object name is omitted (e.g., **AcquisitionStart**() instead of **Camera.AcquisitionStart())** and the default state of the camera is assumed (e.g., Ready for a continuous acquisition start without trigger).

15.1 Acquisition and Trigger examples

/* Continuous acquisition when the camera is in its reset state. */

```
AcquisitionMode = Continuous;
AcquisitionStart();
…
AcquisitionStop();
```

/* Single Frame acquisition in Hardware trigger mode using the external I/O Line 3. */

```
AcquisitionMode    = SingleFrame;
TriggerSelector    = FrameStart;
TriggerMode        = On;
TriggerActivation  = RisingEdge;
TriggerSource      = Line3;
AcquisitionStart();
```

/* Multi-Frames acquisition started by a single Software trigger delayed by
 1 millisecond. The Trigger starts the whole sequence acquisition.
 The Exposure time for each frame is set to 500 us.
*/

```
AcquisitionMode       = MultiFrame;
AcquisitionFrameCount = 20;
TriggerSelector       = AcquisitionStart;
TriggerMode           = On;
TriggerSource         = Software;
TriggerDelayAbs       = 1000;
ExposureMode          = Timed;
ExposureTimeAbs       = 500;
AcquisitionStart();
TriggerSoftware();
```

/* Continuous acquisition in Hardware trigger mode. The Frame triggers are Rising Edge signals coming from the physical Line 2. The Exposure time is 500us. A GigE Vision Event is also sent to the Host application after the exposure of each frame to signal that the inspected part can be moved.
*/

```
AcquisitionMode    = Continuous;
TriggerSelector    = FrameStart;
TriggerMode        = On;
TriggerActivation  = RisingEdge;
TriggerSource      = Line2;
ExposureMode       = Timed;
ExposureTimeAbs    = 500;
EventSelector      = ExposureEnd;
EventNotification  = GigEVisionEvent;
AcquisitionStart();
…
AcquisitionStop();
```

/* Multi-Frames acquisition with each frame triggered by a Hardware trigger on Line 1. A negative pulse of the exposure signal duration (500us) is also sent to the physical output line 2 to activate a light during the exposure time of each frame. The end of the sequence capture is signalled to the host with an acquisition end GigEVision event.

*/

```
AcquisitionMode       = MultiFrame;
AcquisitionFrameCount = 20;
TriggerSelector       = FrameStart;
TriggerMode           = On;
TriggerActivation     = RisingEdge;
TriggerSource         = Line1;
ExposureMode          = Timed;
ExposureTimeAbs       = 500;
LineSelector          = Line2;
LineMode              = Output;
LineInverter          = True;
LineSource            = ExposureActive
EventSelector         = AcquisitionEnd;
EventNotification     = GigEVisionEvent;
AcquisitionStart();
```

/* Line Scan continuous acquisition with Hardware Frame and Line trigger. */

```
AcquisitionMode     = Continuous;
TriggerSelector     = FrameStart;
TriggerMode         = On;
TriggerActivation   = RisingEdge;
TriggerSource       = Line1;
TriggerSelector     = LineStart;
TriggerMode         = On;
TriggerActivation   = RisingEdge;
TriggerSource       = Line2;
AcquisitionStart();
...
AcquisitionStop();
```

/* Frame Scan continuous acquisition with Hardware Frame trigger and the
Exposure duration controlled by the Trigger pulse width.

*/

```
AcquisitionMode     = Continuous;
TriggerSelector     = FrameStart;
TriggerMode         = On;
TriggerActivation   = RisingEdge;
TriggerSource       = Line1;
ExposureMode        = TriggerWidth;
AcquisitionStart();
...
AcquisitionStop();
```

/* Frame Scan continuous acquisition with 1 Hardware trigger controlling
the start of the acquisition and 2 others harware triggers to start and stop
the exposure of each frame.
*/

```
AcquisitionMode = Continuous;
TriggerSelector = AcquistionStart;
TriggerMode     = On;
TriggerSource   = Line1;
ExposureMode    = TriggerControlled;
TriggerSelector = ExposureStart;
TriggerMode     = On;
TriggerSource   = Line3;
TriggerSelector = ExposureStop;
TriggerMode     = On;
TriggerSource   = Line4;
AcquisitionStart();
...
AcquisitionStop();
```

15.2 Counter and Timer examples

/* Counts the number of Triggers received and the number of Frame Start events in a Hardware triggered Continuous acquisition to verify that none were missed.

*/

```
AcquisitionMode    = Continuous;
TriggerSelector    = FrameStart;
TriggerMode        = On;
TriggerActivation  = RisingEdge;
TriggerSource      = Line1;
CounterSelector    = Counter1;
CounterEventSource = FrameTrigger;
CounterReset();
CounterSelector    = Counter2;
CounterEventSource = FrameStart;
CounterReset();
AcquisitionStart();
…
AcquisitionStop();
CounterSelector    = Counter1;
NbTriggers         = CounterValue;
CounterSelector    = Counter2;
NbFrames           = CounterValue;
if (NbTriggers != NbFrames)
    printf(“Error ! Trigger missed.”);
```

/* Use a counter to generate an event at line 200 of each captured Frame in a continuous acquisition.

*/

```
AcquisitionMode      = Continuous;
CounterSelector      = Counter1;
CounterEventSource   = LineStart;
CounterDuration      = 200;
CounterTriggerSource = FrameStart;
EventSelector        = Counter1End;
EventNotification    = GigEVisionEvent;
AcquisitionStart();
…
AcquisitionStop();
```

/* Generate a 200us Timer pulse (Strobe) delayed by 100 us on the physical output
Line 2. The Timer pulse is started using a trigger coming from physical input Line 1.
*/

```
TimerSelector          = Timer1;
TimerDurationAbs       = 200;
TimerDelayAbs          = 100;
TimerTriggerSource     = Line1;
TimerTriggerActivation = RisingEdge;
LineSelector           = Line2;
LineSource             = Timer1Output;
```

/* Use of a Timer to measure the length in microseconds of a negative pulse on the
physical input Line1. An Event is also generated to the host application to
signal the end of the pulse.
*/

```
TimerSelector          = Timer1;
TimerTrigger           = Line1;
TimerTriggerActivation = LevelLow;
EventSelector          = Line1RisingEdge;
EventNotifications     = GigEVisionEvent;
/* Wait for the event on the host to read the time. */
…
TimerSelector          = Timer1;
PulseDuration          = TimerValueAbs;
```

15.3 I/O examples

/* User input of the inverted Status of the physical Line 1. */

```
LineSelector  = Line1;
LineMode      = Input;
LineInverter  = True;
CurrentState  = LineStatus;
```

/* Output of the Exposure signal of each frame on the physical Line 2. */

```
LineSelector = Line2;
LineMode     = Output;
LineSource   = ExposureActive;
```

/* User Output of a positive TTL signal on physical Line 2. */

```
LineSelector        = Line2;
LineMode            = Output;
LineFormat          = TTL;
LineSource          = UserOutput2;
UserOutputSelector  = UserOutput2;
UserOutputValue     = True;
```

16 Acknowledgements

The following companies have participated in the elaboration of the Standard Features Naming Convention:

Company	Represented by
SVS-Vistek	Gerd Reichle
Stemmer Imaging	Sascha Dorenbeck, Rupert Stelz, Martin Kersting
SICK IVP	Mattias Johannesson
Pleora Technologies	Francois Gobeil, Vincent Rowley
National Instruments	Chris Graf, Tony Iglesias, Johann Scholtz, Eric Gross
MVTec Software	Christoph Zierl
Matrox Imaging	Stephane Maurice
Leutron Vision	Jan Becvar, Stefan Thommen
JAI	Tue Moerk, Michael Krag, Ole Krogh Joergensen, Karsten Ingeman Christensen
Euresys	Jean-Michel Wintgens, Yves Joskin
e2v semiconductors	Frédéric Mathieu
DALSA	Peifang Zhou
DALSA Coreco	Eric Carey
Basler	Fritz Dierks, Alex Happe, Andreas Gaer, Thies Möller, Sven Seeger
AIA	Jeff Fryman