GenICam Standard Features Naming Convention

Version 2.1

Table of Content

Table of Figures

History

Version	Date	Changed by	Change
Draft 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)
Draft 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 chapter. 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 UserOutput0 as standard optional 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 GevMACAddress, 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.
Draft 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.
Draft 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
Draft 1.1.01	10.09.2007	Thies Möller, Basler	Created chapter for File Access.
Draft 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.
Draft 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.
Draft 1.2.12	28.10.2008	Stephane Maurice, Matrox Thies Möller, Basler	Matrox: Created draft for 1.3 including: minors corrections, deprecated Raw and Abs feature and deprecated GigEVision Event, Changed chapters names and created according category features, added Root, Device, TLParamsLocked, PixelClock, Temperature features and made ICommand optionally readable, … Basler: Action command was added.
Draft 1.2.13	05.05.2009	Stephane Maurice, Matrox	Deprecated all the GEVSupported… features to regroup them in a selector. Added Color Transformation features. Action command reworked and moved in a separate chapter. Added Event data delivery features.

Version	Date	Changed by	Change
Draft 1.2.14	20.05.2009	Stephane Maurice, Matrox	Deprecated Line0RisingEdge, …compound enumeration in CountersEventSource and created separate CounterEventActivation and CounterResetActivation features to be consistent with the trigger features. Made CounterValue and TimerValue Writable. Modified descriptions to be able to extract tooltips and descriptions for the reference SFNC XML. Added a VBA macro to be able to generate machine readable version of the SFNC. Added a VBA macro to be able to generate the Features summary (Chapter 2) automatically. Changed units to have a standard notation.
Release 1.3	11.08.2009	Stephane Maurice, Matrox	SFNC 1.3 release including the changes since version 1.2.1.
Draft 1.4	05.01.2009 and 22.01.2010	Vincent Rowley, Pleora Technologies Inc.	Added GigE Vision 1.2 support. Added missing Bpp36 and Bpp48 enumeration entries for PixelSize feature. Added missing RawPacked enumeration entry for PixelCoding feature. Updated support level for GevSCPInterfaceIndex feature in order to be consistent with related features. Clarified text when necessary and fixed typos. Corrected some feature descriptions.

Version	Date	Changed by	Change
Release 1.4	17.03.2010	Stephane Maurice, Matrox	Minor fixes to remove mistakes. YUV422YUYVPacked was removed, changed all the ExposureTimeAuto to ExposureAuto. Corrected GevGVCPPendingAck and GevManifestSecondaryURL names. Added ChunkTimer and ChunkCounter to ChunkSelector. Updated VB macros.
Release 1.5	22.11.2010	Stephane Maurice, Matrox	- Added Camera Link related features. - ActionSelector now >0. - Added Bpp30 to PixelSize. -Added RGB16Packed, BGR16Packed, BGR10V1Packed and BGR10V2Packed to pixel Format. - DeviceUserID is now recommended to be an empty string. - GenICam Access chapter added. - DeviceSFNCVersion… features added. - Clarified and corrected points in Counter and Timer chapter. - Added new TimerReset feature and removed LevelHigh and LevelLow in CounterEventActivation of Counter and Timer chapter. - Updated Chunk chapter to correct inconsistencies and add missing items. - Minor fixes to remove mistakes. - Updated VB macros to extract the description of the enumerations and fix other minor parsing issues.

Version	Date	Changed by	Change
Release 1.5.1	20.09.2011	Stephane Maurice, Matrox	- Added FrameBurst triggers functionality and their associated features. - Corrected signal names in the figures of the Acquisition control chapter. - Added acquisition timing diagrams in the Acquisition control chapter to illustrate the typical acquisition cases. - Removed and changed some GigE Vision mentions that were too standard specific to accommodate other TL standards. - UserOutputValueAll now mentions that UserOutput0 maps to the Lsb of the register instead of saying Bit 0. - AcquisitionStart is now (Read)/Write instead of Read/Write. - Added Timestamp features. - Added clarifications for Width, WidthMax, Height, and HeightMax. - Added clarifications for the usage of the All enumeration of GainSelector, BlackLevelSelector and WhiteClipSelector. - Replaced AOI per ROI to better match other standards nomenclature. - Made DeviceTapGeometry less Camera Link specific. - Added a note for ExposureTime that the feature is inactive if ExposureAuto is On. - Added a note for AcquisitionLineRate that the feature is inactive if TriggerMode is On. - Minor corrections from 1.5.

Version	Date	Changed by	Change
Draft 1 1.5.2	23.12.2011	Stephane Maurice, Matrox	- Moved the document to Word .Docm format (.Docx with macro). - Removed and changed some GigE Vision references that were too standard specific to accommodate other TL standards. - Generalized many GEV features that are now in Device Control chapter to be usable by other TL. - Added DeviceLinkThroughputLimit features. - Updated the PixelFormat and ChunkPixelFormat features and the corresponding text according to the new Pixel Format Naming Convention of the AIA. - Added a standard definitions table and figure.
Draft 2 1.5.2	23.01.2012	Vincent Rowley, Pleora Technologies Inc.	- Added GigE Vision 2.0 support. o Added ActionLate event. o Added support for link aggregation. o Added support for PAUSE frames. o Added support for extended and standard IDs modes. o Added support for IEEE 1588. o Added support for unconditional and scheduled action commands. o Added support for extended status codes introduced by GigE Vision 2.0. o Added multi-zone support. o Added support for alternate packet resend destination option. o Added support for All-in Transmission mode. - Added PrimaryApplicationSwitch and LinkSpeedChange events. - Added support for 10-tap geometries.

Version	Date	Changed by	Change
Release 2.0	25.10.2012	Stephane Maurice, Matrox	- Include all items from 1.5.2 Drafts - Changed all the Mandatory features that were not GenICam related to Recommended to let each TL standards define this. - Replaced "Recommended Visibility" by "Visibility" only. - Added notes to mention that Visibility and Category are recommended. - Deprecated PixelCoding. - Deprecated DeviceID and replaced it with DeviceSerialNumber. - Deprecated TestImageSelector and replaced it with TestPattern. - Deprecated UserSetDefaultSelector and replaced it with UserSetDefault. - Removed all the features and enumerations that were deprecated before this major version of the SFNC. - Added the Device Communication and Device Acquisition Model section. - Added the multiple Region (ROI) handling features (RegionSelector, ...) to the ImageFormat Control chapter. - Added the multiple Source handling features (SourceSelector, ...). - Added the Stream Transfer handling features (TransferSelector, ...). - Added a note for a possible optional DeviceStreamChannelSelector for PayloadSize . - Updated the Selector description section.

Version	Date	Changed by	Change
Release 2.0 (cont.)	25.10.2012	Stephane Maurice, Matrox	- Added option to have 0 based Timers, Counters, Actions, Sources, Streams,… - Added the CoaXPress transport layer specific features. - Added the Deinterlacing feature. - Added the Image Compression features. - Added DeviceIndicatorMode feature. - GenICam Access chapter was renamed GenICam Control. - Updated the VB macro.
Release 2.0 (syntax update)	30.10.2012	Stephane Maurice, Matrox	- Minor correction to the text format of few features to help the parsing by theVB macro. - Updated the VB macro.
Release 2.1	12.12.2013	Stephane Maurice, Matrox	- Added UserSetFeatureSelector and UserSetFeatureEnable features. - Added Sequencer Control category and other sequencer related features. - Added Software Signal Control category and other related features. - Added Geometry_4X2E and Geometry_4X2E_1Y tap geometries. - Deprecated the Gev… features that were redundant with the Device… features (Ex: GevDeviceClass is replaced by the generic DeviceType). - Added PixelFormatInfoSelector and PixelFormatInfoId features. - Update to the CXP TL specific features description. - Allowed PayloadSize to be 0. - Created TimestampLatch, TimestampLatchValue and ChunkTimestampLatchValue features and deprecated their GigE Vision

			counterpart. - Deprecated DeviceMessageChannelCount and replaced it with DeviceEventChannelCount. The word Message was also replaced with Event everywhere in the text to be consistent with the Event features name. - Fixed DeviceStreamChannelEndianess by changing it to IEnumeration instead of IBoolean. - Minor corrections to the text. - Updated the VB macro.

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 the features of a device

The usage of GenApi alone could be sufficient to make all the features of a camera or a device accessible through the GenICam API. However if the user wants to write **generic and portable software** for a whole class of cameras or devices and be interoperable, then GenApi alone is not sufficient and the software and the device vendors have to agree on a common naming convention for the standard features. This is the role of the GenICam "Standard Features Naming Convention (SFNC)" to provide this common set of features, their name, and to define a standard behavior for them.

The Standard Features Naming Convention of GenICam is targeting maximum usability by existing and future transport layer technologies. It provides the definitions of a **standard behavioral model** and of **standard features**. The goal is to cover and to standardize the naming convention used in all the basic use cases where the implementation by different vendors would be very similar anyway.

Note that to be GenICam compliant a product:

- Must provide or accept a GenICam XML file compatible with the latest GenApi and schema.
- The GenICam XML file must include all the public features of the product it describes.
- The GenICam XML must follow the Standard Features Naming Convention whenever applicable or possible for its features.

Those requirements ensure that the users can rely on a complete, consistent and portable feature set for its device and that those features are always accessible in a standard way.

1.1 Conventions

Feature Name and Interface

According to the GenICam standard, all the public features of a device must be included in the GenICam XML file and must use the SFNC Name and Interface type for those features if they exist. Other vendor specifc or specialized features not mapping to existing SNFC features can be included but must be located in a vendor specific namespace in the GenICam XML and may use a vendor specific name.

This document lists for each feature, the Name and Interface type that must be used.

Feature Category

With the GenICam standard, each feature should be included in a "Category". The Category element defines in which group of features, the feature will be located.

The Category does not affect the functionality of the features but is used by the GUIs to group the features when displaying them. The purpose is mainly to insure that the GUI can present features in a more organized way.

This document lists for each feature, a recommended Category that should be used.

Feature Level

In this document, features are tagged according to the following requirement levels:

- M: **Mandatory -** Must be implemented to achieve compliance with the GenICam standard.
- R: Recommended **-** This feature adds important aspects to the use case and must respect the naming convention if used.
- O: Optional - This feature is less critical. Nevertheless, it is considered and must respect the naming convention if used.

For additional details about the mandatory features specific to a particular transport layers, please refer to the text of those standards.

Feature Visibility

According to the GenICam standard each feature can be assigned a "Visibility". The Visibility defines the type of user that should get access to the feature. Possible values are: Beginner, Expert, Guru and Invisible. The latter is required to make features accessible from the API, but invisible in the GUI.

The visibility does not affect the functionality of the features but is 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 have been used for the assignment of the 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 XMLfiles and will be used if the Visibility element is omitted for a feature. The number of features with "Beginner" visibility should be limited to all **basic** features of the devices so the GUI display is well-organized 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.

This document lists for each feature, a recommended Visibility that should be used.

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 of the red/green/blue component of a color camera).

A selector is a separate feature that is typically an IEnumeration or an IInteger. 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.

If a selector has only one possible value, the selector relation can be omitted but it is recommended to leave the selector feature as read only for information purpose (Ex: TriggerSelector = FrameStart (read only) for a device that has only this trigger type supported).

In this document, the features potentially dependent on a 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.

In general, a selector should apply only to a single category of feature (Ex: TriggerSelector applies only to the Trigger related features). However, it is possible that certain more advanced devices will require a selector that applies to features in different categories. For example a device with 2 independent input sensors could have a SourceSelector feature that would select features in the Image Format Control, Acquisition Control, Analog Control, LUT Control and Color Transformation Control categories in order to globally control all the features associated with a particular source (Ex": SourceSelector = Source1, PixelFormat[SourceSelector] = Mono8, Gain[SourceSelector] = 10, AcquisitionStart[SourceSelector]).

Note also that when a feature that has a selector is persisted to a file, the selector is iterated to allow saving the complete array of values and not only the currently selected element.

1.2 Standard Units

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

ns	nanoseconds
us	microseconds
ms	milliseconds
s	seconds
B	Bytes
Bps	Bytes per second
MBps	Mega Bytes per second
Mbps	Mega bits per second
Fps	Frames per second
dB	decibels
C	Celsius
Hz	Hertz

1.3 Acronyms

The following definitions are used in this document.

ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AIA	Automated Imaging Association
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

1.4 Standard Definitions

This section defines the terms used in this document. An illustration of their inter-relation is provided in the

Device Communication Model section below (see Figure 1-1).

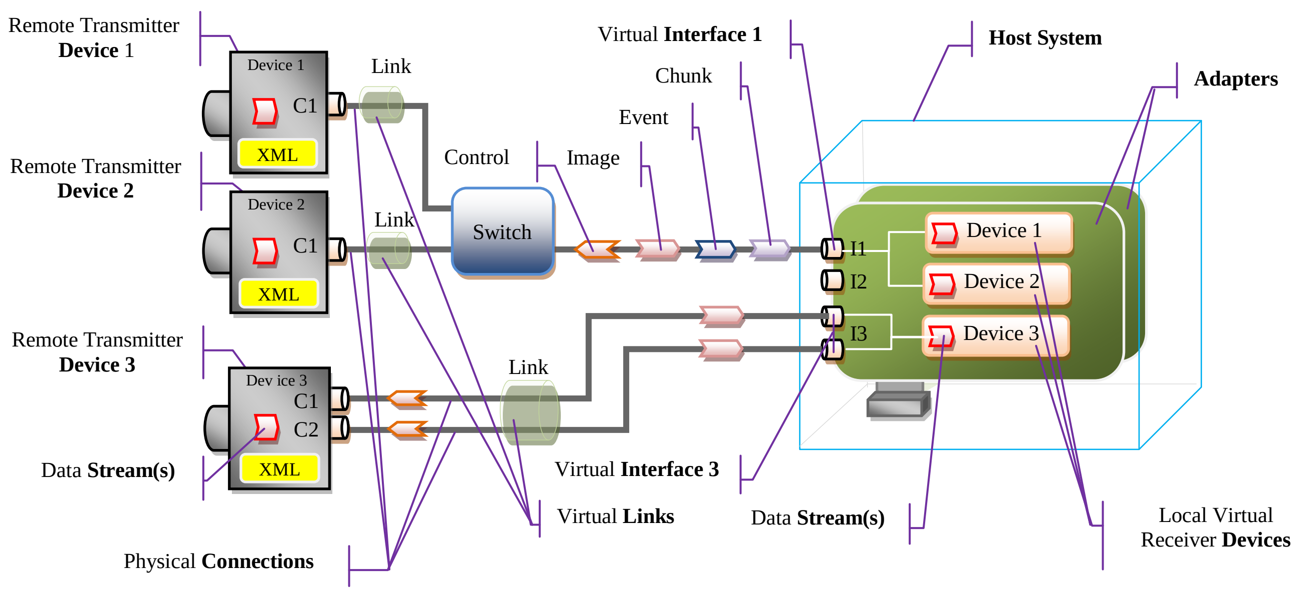
Entity	An *Entity* is an end point located at either side (*Host* or *Device*) of a *Communication*.
Host System	The *Host System* is the *Entity* which takes control over a *Device*. A *Host System* can be the sink or the source for the data being streamed. Under GenICam the *Host System* must read and use the GenICam compliant XML file of the *Device* to control it.
Device	The *Device* is an *Entity* which is controlled by a *Host System*. A *Device* can be the source or the sink for streaming data. It can be remote (outside the *Host System*) or local (in the *Host System*). Under GenICam the *Device* must provide a GenICam compliant XML file and a register-based control access.
Link	A *Link* is the virtual binding between a *Host System* and a *Device* to establish a *Communication*. A *Link* is logical and may use one or more physical *Connections*.
Connection	A *Connection* is the physical binding between a *Host System* and a *Device*.
Interface	A virtual end point of the *Link* between a *Device* and a *Host System*.
Adapter	A physical entity located in the Host System that has one or many *Interfaces.*
Communication	A *Communication* is an exchange of information between two *Entities* using a *Link*.
Channel	A logical point-to-point *Communication* over a *Link*. There may be multiple *Channels* on a single *Link*.
Transport Layer	The layer of *Communication* responsible to transport information between *Entities*.
Transmitter	An *Entity* which acts as the source for streaming data. This may apply to a *Host System* or a *Device*.
Receiver	An *Entity* which acts as the sink for streaming data. This may apply to a *Host System* or a *Device*.
Transceiver	An *Entity* which can receive and transmit streaming data. This may apply to a *Host System* or a *Device*.
Peripheral	An *Entity* which neither acts as a source nor as a sink for streaming data but can be controlled.
Stream	A flow of data that comes from a source and goes to a sink. A data *Stream* can be composed of images or chunk of data.
Stream Channel	A *Communication Channel* used to transmit a data *Stream* from a *Transmitter* (or *Transceiver*) to a *Receiver* (or *Transceiver*).
Event Channel	A *Communication Channel* used by the *Device* to notify the *Host System* asynchronously of *Events*. The *Host System* could also use a *Event Channel* to communicate events to the *Device.*

Control Channel	A *Communication Channel* used to configure and control a *Device*. For a *Control Channel* the *Device* acts as a server that provides the initial point of *Communication* for the *Host System* that acts as a Client. The *Communication* on a *Control Channel* is bidirectional and initiated by the *Host System*.
Event	An asynchronous notification of the occurrence of a fact. *Events* are transmitted on an *Event Channel*.

1.5 Device Communication Model

This section presents the general communication model for the devices controlled using the SFNC. It presents the main elements involved in the communication for control and data streaming between the Host System and the acquisition Device.

Figure 1-1 Device Communication Model



In general, the Device Communication model is:

The remote Device and the Host System communicate using a virtual Link.
The virtual Link is established on an Interface using one or more physical Connections.
The Host System controls the remote Device using the features present in its GenICam XML file.
The remote Transmitter Device has a data Source that generates a data Stream.
The data Stream is sent to the Host System on a Stream Channel of the virtual Link.
The reception of the data Stream on a Host Interface is handled by a local receiver Device.
The local receiver Device writes the data Stream to the Host System memory.

See section 1.4 Standard Definitions section above for more detailed information.

1.6 Device Acquisition Model

This section presents the general data acquisition model for the devices controlled using the SFNC. It presents the main elements involved in the data acquisition by a Device and the typical data flow for transfer of images to the Host System. It covers the typical devices with a single data source and the more complex devices with multi-source, multi-region of interest and data transfer control.

Basic acquisition Devices with one source of data, one region of interest and automatic control of the transfer of data such as the one shown in Figure 1-2, are simple particular case of this model where the Source, Region and Transfer features are fixed and cannot be changed (so the corresponding fixed features can be omitted).

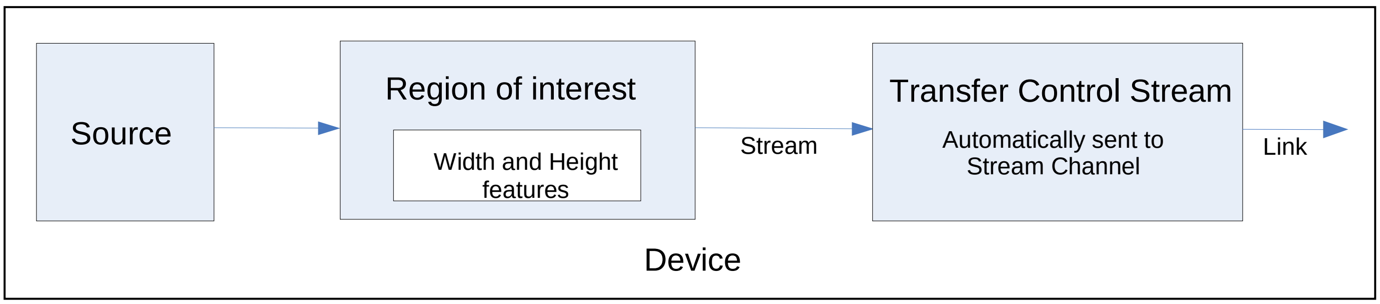


Figure 1-2: Basic acquisition device with fixed configuration.

The typical features setting for such a basic device where the values of Source, Region and Transfer Control Stream are fixed is typically reduced to:

Width = 320
Height = 240
AcquisitionStart
…
AcquisitionStop

But in general, for more complex devices, the acquisition and data transfer model is:

A Device has one or many Source(s).
A Source has one or many Region(s) of interest.
A Region of interest goes to a data Stream.
The data generation by the Source is controlled by the "Acquisition Control" features.
The dimensions of a Region of interest are controlled by the "Image Format Control" features.
The outgoing data flow of a Stream is controlled by a "Transfer Control" features.
The Transfer Control module output the data Stream to a Stream Channel.
The Stream Channel is transmitted on a virtual Link.
The virtual Link is established with a Host System using one or many Device`s physical Connection(s).

Figure 1-3 below presents an example of a more complex device supporting multi-source, multi-region with data stream transfer control.

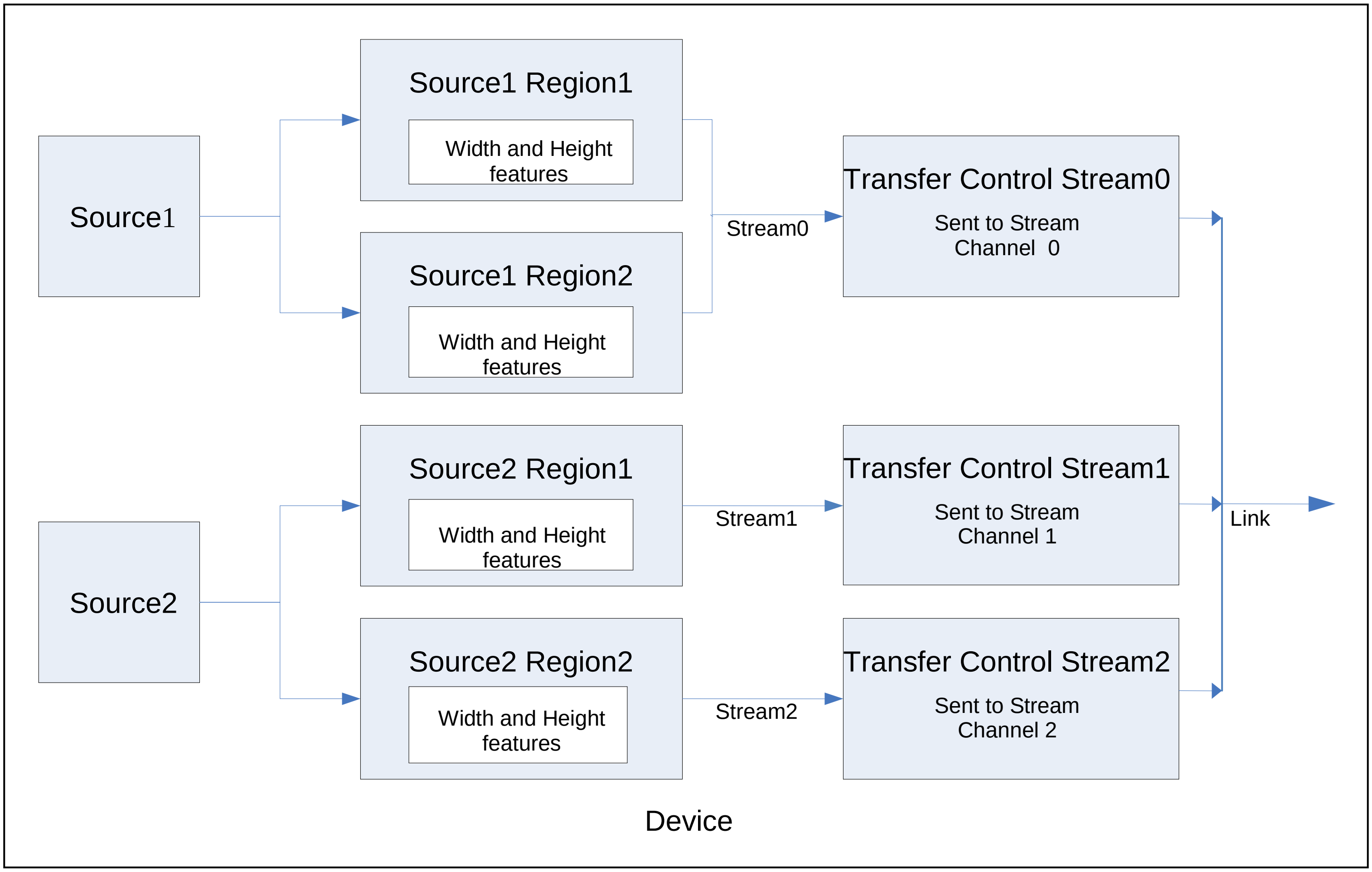


Figure 1-3: Multi-source, multi-region acquisition device with data stream transfer control.

The typical feature setting model for such a complex multi-source device is presented in detail the chapter 18 (Source Control). The features for the control of the regions of interest and image format handling are documented in chapter 4 (Image Format Control) and Chapter 19 (Transfer Control) presents the features to control the flow of data on the external link.

2 Features Summary

This chapter provides a comprehensive summary of the standard features covered by this document. The following chapters provide more detailed explanation of each feature.

In case of discrepancy, the sections describing the features in detail prevail.

2.1 Device Control

Contains the features related to the control and information of the device.

Table 2-1: Device Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
DeviceControl	R	ICategory	R	-	B	Category for device information and control.
DeviceType	O	IEnumeration	R	-	G	Returns the device type.
DeviceScanType	R	IEnumeration	R/(W)	-	E	Scan type of the sensor of the device.
DeviceVendorName	R	IString	R	-	B	Name of the manufacturer of the device.
DeviceModelName	R	IString	R	-	B	Model of the device.
DeviceFamilyName	O	IString	R	-	B	Identifier of the product family of the device.
DeviceManufacturerInfo	R	IString	R	-	B	Manufacturer information about the device.
DeviceVersion	R	IString	R	-	B	Version of the device.
DeviceFirmwareVersion	R	IString	R	-	B	Version of the firmware in the device.
DeviceSerialNumber	R	IString	R	-	E	Device`s serial number.
DeviceID	R	IString	R	-	I	This feature is deprecated (See DeviceSerialNumber).
DeviceUserID	O	IString	R/W	-	B	User-programmable device identifier.
DeviceSFNCVersionMajor	R	IInteger	R	-	B	Major version of the Standard Features Naming Convention that was used to create the device`s GenICam XML.

DeviceSFNCVersionMinor	R	IInteger	R	-	B	Minor version of the Standard Features Naming Convention that was used to create the device`s GenICam XML.
DeviceSFNCVersionSubMinor	R	IInteger	R	-	B	Sub minor version of Standard Features Naming Convention that was used to create the device`s GenICam XML.
DeviceManifestEntrySelector	O	IInteger	R/W	-	G	Selects the manifest entry to reference.
DeviceManifestXMLMajorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the major version number of the GenICam XML file of the selected manifest entry.
DeviceManifestXMLMinorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the minor version number of the GenICam XML file of the selected manifest entry.
DeviceManifestXMLSubMinorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the subminor version number of the GenICam XML file of the selected manifest entry.
DeviceManifestSchemaMajorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the major version number of the schema file of the selected manifest entry.
DeviceManifestSchemaMinorVersion[DeviceManifestEntrySelector]	O	IInteger	R	-	G	Indicates the minor version number of the schema file of the selected manifest entry.
DeviceManifestPrimaryURL[DeviceManifestEntrySelector]	O	IString	R	-	G	Indicates the first URL to the GenICam XML device description file of the selected manifest entry.
DeviceManifestSecondaryURL[DeviceManifestEntrySelector]	O	IString	R	-	G	Indicates the second URL to the GenICam XML device description file of the selected manifest entry.
DeviceTLType	R	IEnumeration	R	-	B	Transport Layer type of the device.
DeviceTLVersionMajor	R	IInteger	R	-	B	Major version of the Transport Layer of the device.
DeviceTLVersionMinor	R	IInteger	R	-	B	Minor version of the Transport Layer of the device.
DeviceTLVersionSubMinor	R	IInteger	R	-	B	Sub minor version of the Transport Layer of the device.
DeviceGenCPVersionMajor	R	IInteger	R	-	B	Major version of the GenCP protocol supported by the device.
DeviceGenCPVersionMinor	R	IInteger	R	-	B	Minor version of the GenCP protocol supported by the device.
DeviceMaxThroughput	O	IInteger	R	Bps	E	Maximum bandwidth of the data that can be streamed out of the device.
DeviceConnectionSelector	R	IInteger	R/(W)	-	B	Selects which Connection of the device to control.
DeviceConnectionSpeed[DeviceConnect	O	IInteger	R	Bps	E	Indicates the speed of transmission of the specified Connection.

ionSelector]						
DeviceConnectionStatus[DeviceConnectionSelector]	O	IEnumeration	R	-	E	Indicates the status of the specified Connection.
DeviceLinkSelector	R	IInteger	R/(W)	-	B	Selects which Link of the device to control.
DeviceLinkSpeed[DeviceLinkSelector]	O	IInteger	R	Bps	E	Indicates the speed of transmission negotiated on the specified Link.
DeviceLinkThroughputLimitMode[DeviceLinkSelector]	R	IEnumeration	R/W	-	E	Controls if the DeviceLinkThroughputLimit is active.
DeviceLinkThroughputLimit[DeviceLinkSelector]	R	IInteger	R/(W)	Bps	E	Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.
DeviceLinkConnectionCount[DeviceLinkSelector]	O	IInteger	R	-	B	Returns the number of physical connection of the device used by a particular Link.
DeviceLinkHeartbeatMode[DeviceLinkSelector]	O	IEnumeration	R/W	-	E	Activate or deactivate the Link`s heartbeat.
DeviceLinkHeartbeatTimeout[DeviceLinkSelector]	O	IFloat	R/W	us	G	Controls the current heartbeat timeout of the specific Link.
DeviceCommandTimeout [DeviceLinkSelector]	O	IFloat	R	us	G	Indicates the command timeout of the specified Link.
DeviceStreamChannelCount	O	IInteger	R	-	E	Indicates the number of streaming channels supported by the device.
DeviceStreamChannelSelector	O	IInteger	R/W	-	E	Selects the stream channel to control.
DeviceStreamChannelType[DeviceStreamChannelSelector]	O	IEnumeration	R	-	G	Reports the type of the stream channel.
DeviceStreamChannelLink[DeviceStreamChannelSelector]	O	IInteger	R/(W)	-	G	Index of device`s Link to use for streaming the specifed stream channel.
DeviceStreamChannelEndianness [DeviceStreamChannelSelector]	O	IEnumeration	R/(W)	-	G	Endianess of multi-byte pixel data for this stream.
DeviceStreamChannelPacketSize [DeviceStreamChannelSelector]	R	IInteger	R/(W)	B	E	Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.
DeviceEventChannelCount	O	IInteger	R	-	E	Indicates the number of event channels supported by the device.

DeviceMessageChannelCount	O	IInteger	R	-	E	This feature is deprecated (See DeviceEventChannelCount).
DeviceCharacterSet	O	IEnumeration	R	-	G	Character set used by the strings of the device`s bootstrap registers.
DeviceReset	R	ICommand	W	-	G	Resets the device to its power up state.
DeviceIndicatorMode	O	IEnumeration	R/W	-	E	Controls the behavior of the indicators (such as LEDs) showing the status of the Device.
DeviceFeaturePersistenceStart	O	ICommand	(R)/W	-	G	Indicate to the device and GenICam XML to get ready for persisting of all streamable features.
DeviceFeaturePersistenceEnd	O	ICommand	(R)/W	-	G	Indicate to the device the end of feature persistence.
DeviceRegistersStreamingStart	R	ICommand	(R)/W	-	G	Prepare the device for registers streaming without checking for consistency.
DeviceRegistersStreamingEnd	R	ICommand	(R)/W	-	G	Announce the end of registers streaming.
DeviceRegistersCheck	R	ICommand	(R)/W	-	E	Perform the validation of the current register set for consistency.
DeviceRegistersValid	R	IBoolean	R	-	E	Returns if the current register set is valid and consistent.
DeviceRegistersEndianness	O	IEnumeration	R/(W)	-	G	Endianess of the registers of the device.
DeviceTemperatureSelector	O	IEnumeration	R/W	-	E	Selects the location within the device, where the temperature will be measured.
DeviceTemperature[DeviceTemperatureSelector]	O	IFloat	R	C	E	Device temperature in degrees Celsius (C).
DeviceClockSelector	O	IEnumeration	R/(W)	-	E	Selects the clock frequency to access from the device.
DeviceClockFrequency[DeviceClockSelector]	O	IFloat	R/(W)	Hz	E	Returns the frequency of the selected Clock.
DeviceSerialPortSelector	R	IEnumeration	R/(W)	-	E	Selects which serial port of the device to control.
DeviceSerialPortBaudRate[DeviceSerialPortSelector]	R	IEnumeration	R/(W)	-	E	This feature controls the baud rate used by the selected serial port.
Timestamp	R	IInteger	R	ns	E	Reports the current value of the device timestamp counter.
TimestampReset	O	ICommand	(R)/W	-	E	Resets the current value of the device timestamp counter.
TimestampLatch	O	ICommand	W	-	E	Latches the current timestamp counter into TimestampLatchValue.

TimestampLatchValue	O	IInteger	R	ns	E	Returns the latched value of the timestamp counter.

2.2 Image Format Control

Contains the features related to the format of the transmitted image.

Table 2-2: Image Format Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
ImageFormatControl	R	ICategory	R	-	B	Category for Image Format Control features.
SensorWidth	R	IInteger	R	-	E	Effective width of the sensor in pixels.
SensorHeight	R	IInteger	R	-	E	Effective height of the sensor in 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	-	E	Maximum width of the image (in pixels).
HeightMax	R	IInteger	R	-	E	Maximum height of the image (in pixels).
RegionSelector	O	IEnumeration	R/(W)	-	B	Selects the Region of interest to control.
RegionMode[RegionSelector]	O	IEnumeration	R/W	-	B	Controls if the selected Region of interest is active and streaming.
RegionDestination[RegionSelector]	O	IEnumeration	R/(W)	-	E	Control the destination of the selected region.
Width[RegionSelector]	R	IInteger	R/(W)	-	B	Width of the image provided by the device (in pixels).
Height[RegionSelector]	R	IInteger	R/(W)	-	B	Height of the image provided by the device (in pixels).
OffsetX[RegionSelector]	R	IInteger	R/W	-	B	Horizontal offset from the origin to the region of interest (in pixels).
OffsetY[RegionSelector]	R	IInteger	R/W	-	B	Vertical offset from the origin to the region of interest (in pixels).
LinePitch[RegionSelector]	R	IInteger	R/W	B	E	Total number of bytes between 2 successive lines.
BinningSelector	O	IEnumeration	R/(W)	-	E	Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

BinningHorizontal[BinningSelector]	O	IInteger	R/W	-	E	Number of horizontal photo-sensitive cells to combine together.
BinningVertical[BinningSelector]	O	IInteger	R/W	-	E	Number of vertical photo-sensitive cells to combine together.
DecimationHorizontal	O	IInteger	R/W	-	E	Horizontal sub-sampling of the image.
DecimationVertical	O	IInteger	R/W	-	E	Vertical sub-sampling of the image.
ReverseX	R	IBoolean	R/W	-	E	Flip horizontally the image sent by the device.
ReverseY	R	IBoolean	R/W	-	E	Flip vertically the image sent by the device.
PixelFormat	R	IEnumeration	R/(W)	-	B	Format of the pixels provided by the device.
PixelFormatInfoSelector	R	IEnumeration	R/W	-	G	Select the pixel format for which the information will be returned.
PixelFormatInfoID[PixelFormatInfoSelector]	R	IInteger	R	-	G	Returns the value used by the streaming channels to identify the selected pixel format.
PixelCoding	R	IEnumeration	R/(W)	-	E	This feature is deprecated.
PixelSize	R	IEnumeration	R/(W)	-	E	Total size in bits of a pixel of the image.
PixelColorFilter	R	IEnumeration	R/(W)	-	E	Type of color filter that is applied to the image.
PixelDynamicRangeMin	O	IInteger	R/W	-	E	Minimum value that can be returned during the digitization process.
PixelDynamicRangeMax	O	IInteger	R/W	-	E	Maximum value that will be returned during the digitization process.
TestPatternGeneratorSelector	O	IEnumeration	R/(W)	-	B	Selects which test pattern generator is controlled by the TestPattern feature.
TestPattern[TestPatternGeneratorSelector]	O	IEnumeration	R/W	-	B	Selects the type of test pattern that is generated by the device as image source.
TestImageSelector	O	IEnumeration	R/W	-	I	This feature is deprecated (see TestPattern).
Deinterlacing	O	IEnumeration	R/W	-	B	Controls how the device performs de-interlacing.
ImageCompressionMode	O	IEnumeration	R/W	-	B	Enable a specific image compression mode as the base mode for image transfer.
ImageCompressionRateOption	O	IEnumeration	R/W	-	E	Two rate controlling options are offered: fixed bit rate or fixed quality.
ImageCompressionQuality	O	IInteger	R/(W)	-	E	Control the quality of the produced compressed stream.
ImageCompressionBitrate	O	IFloat	R/(W)	Mbps	E	Control the rate of the produced compressed stream.

ImageCompressionJPEGFormatOption	O	IEnumeration	R/W	-	E	When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

2.3 Acquisition Control

Contains the features related to image acquisition, including the triggering mode.

Table 2-3: Acquisition Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
AcquisitionControl	R	ICategory	R	-	B	Category for the acquisition and trigger control features.
AcquisitionMode	R	IEnumeration	R/(W)	-	B	Sets the acquisition mode of the device.
AcquisitionStart	R	ICommand	(R)/W	-	B	Starts the Acquisition of the device.
AcquisitionStop	R	ICommand	(R)/W	-	B	Stops the Acquisition of the device at the end of the current Frame.
AcquisitionAbort	R	ICommand	(R)/W	-	E	Aborts the Acquisition immediately.
AcquisitionArm	O	ICommand	(R)/W	-	E	Arms the device before an AcquisitionStart command.
AcquisitionFrameCount	R	IInteger	R/W	-	B	Number of frames to acquire in MultiFrame Acquisition mode.
AcquisitionBurstFrameCount	O	IInteger	R/W	-	B	Number of frames to acquire for each FrameBurstStart trigger.
AcquisitionFrameRate	R	IFloat	R/W	Hz	B	Controls the acquisition rate (in Hertz) at which the frames are captured.
AcquisitionLineRate	R	IFloat	R/W	Hz	B	Controls the rate (in Hertz) at which the Lines in a Frame are captured.
AcquisitionStatusSelector	R	IEnumeration	R/W	-	E	Selects the internal acquisition signal to read using AcquisitionStatus.
AcquisitionStatus[AcquisitionStatusSelector]	R	IBoolean	R	-	E	Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.
TriggerSelector	R	IEnumeration	R/W	-	B	Selects the type of trigger to configure.
TriggerMode[TriggerSelector]	R	IEnumeration	R/W	-	B	Controls if the selected trigger is active.
TriggerSoftware[TriggerSelector]	R	ICommand	(R)/W	-	B	Generates an internal trigger.

TriggerSource[TriggerSelector]	R	IEnumeration	R/W	-	B	Specifies the internal signal or physical input Line to use as the trigger source.
TriggerActivation[TriggerSelector]	R	IEnumeration	R/W	-	B	Specifies the activation mode of the trigger.
TriggerOverlap[TriggerSelector]	R	IEnumeration	R/W	-	E	Specifies the type trigger overlap permitted with the previous frame.
TriggerDelay[TriggerSelector]	R	IFloat	R/W	us	E	Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.
TriggerDivider[TriggerSelector]	R	IInteger	R/W	-	E	Specifies a division factor for the incoming trigger pulses.
TriggerMultiplier[TriggerSelector]	R	IInteger	R/W	-	E	Specifies a multiplication factor for the incoming trigger pulses.
ExposureMode	R	IEnumeration	R/W	-	B	Sets the operation mode of the Exposure (or shutter).
ExposureTime	R	IFloat	R/W	us	B	Sets the Exposure time when ExposureMode is Timed and ExposureAuto is Off.
ExposureAuto	O	IEnumeration	R/W	-	B	Sets the automatic exposure mode when ExposureMode is Timed.

2.4 Digital I/O Control

Contains the features related to the control of the general input and output pins of the device.

Table 2-4: Digital I/O Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
DigitalIOControl	R	ICategory	R	-	E	Category that contains the digital input and output control features.
LineSelector	R	IEnumeration	R/W	-	E	Selects the physical line (or pin) of the external device connector to configure.
LineMode[LineSelector]	O	IEnumeration	R/W	-	E	Controls if the physical Line is used to Input or Output a signal.
LineInverter[LineSelector]	R	IBoolean	R/W	-	E	Controls the inversion of the signal of the selected input or output Line.
LineStatus[LineSelector]	R	IBoolean	R	-	E	Returns the current status of the selected input or output Line.
LineStatusAll	O	IInteger	R	-	E	Returns the current status of all available Line signals at time of polling

						in a single bitfield.
LineSource[LineSelector]	R	IEnumeration	R/W	-	E	Selects which internal acquisition or I/O source signal to output on the selected Line.
LineFormat[LineSelector]	O	IEnumeration	R/W	-	E	Controls the current electrical format of the selected physical input or output Line.
UserOutputSelector	R	IEnumeration	R/W	-	E	Selects which bit of the User Output register will be set by UserOutputValue.
UserOutputValue[UserOutputSelector]	R	IBoolean	R/W	-	E	Sets the value of the bit selected by UserOutputSelector.
UserOutputValueAll	O	IInteger	R/W	-	E	Sets the value of all the bits of the User Output register.
UserOutputValueAllMask	O	IInteger	R/W	-	E	Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.

2.5 Counter and Timer Control

Contains the features related to the usage of programmable counters and timers.

Table 2-5: Counter and Timer Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
CounterAndTimerControl	R	ICategory	R	-	E	Category that contains the Counter and Timer control features.
CounterSelector	R	IEnumeration	R/W	-	E	Selects which Counter to configure.
CounterEventSource[CounterSelector]	R	IEnumeration	R/W	-	E	Select the events that will be the source to increment the Counter.
CounterEventActivation[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the Activation mode Event Source signal.
CounterResetSource[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the signals that will be the source to reset the Counter.
CounterResetActivation[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the Activation mode of the Counter Reset Source signal.
CounterReset[CounterSelector]	R	ICommand	(R)/W	-	E	Does a software reset of the selected Counter and starts it.

CounterValue[CounterSelector]	R	IInteger	R/W	-	E	Reads or writes the current value of the selected Counter.
CounterValueAtReset[CounterSelector]	R	IInteger	R	-	E	Reads the value of the selected Counter when it was reset by a trigger or by an explicit CounterReset command.
CounterDuration[CounterSelector]	R	IInteger	R/W	-	E	Sets the duration (or number of events) before the CounterEnd event is generated.
CounterStatus[CounterSelector]	R	IEnumeration	R	-	E	Returns the current status of the Counter.
CounterTriggerSource[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the source to start the Counter.
CounterTriggerActivation[CounterSelector]	R	IEnumeration	R/W	-	E	Selects the activation mode of the trigger to start the Counter.
TimerSelector	R	IEnumeration	R/W	-	E	Selects which Timer to configure.
TimerDuration[TimerSelector]	R	IFloat	R/W	us	E	Sets the duration (in microseconds) of the Timer pulse.
TimerDelay[TimerSelector]	R	IFloat	R/W	us	E	Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.
TimerReset[TimerSelector]	R	ICommand	(R)/W	-	E	Does a software reset of the selected timer and starts it.
TimerValue[TimerSelector]	R	IFloat	R/W	us	E	Reads or writes the current value (in microseconds) of the selected Timer.
TimerStatus[TimerSelector]	R	IEnumeration	R	-	E	Returns the current status of the Timer.
TimerTriggerSource[TimerSelector]	R	IEnumeration	R/W	-	E	Selects the source of the trigger to start the Timer.
TimerTriggerActivation[TimerSelector]	R	IEnumeration	R/W	-	E	Selects the activation mode of the trigger to start the Timer.

2.6 Event Control

Contains the features related to the generation of Event notifications by the device.

Table 2-6: Event Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description

EventControl	R	ICategory	R	-	E	Category that contains Event control features.
EventSelector	R	IEnumeration	R/W	-	E	Selects which Event to signal to the host application.
EventNotification[EventSelector]	R	IEnumeration	R/W	-	E	Activate or deactivate the notification to the host application of the occurrence of the selected Event.

Name	Level	Interface	Access	Unit	Visibility	Description
EventFrameTriggerData	R	ICategory	R	-	E	Category that contains all the data features related to the FrameTrigger Event.
EventFrameTrigger	R	IInteger	R	-	E	Returns the unique Identifier of the FrameTrigger type of Event.
EventFrameTriggerTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the AcquisitionTrigger Event.
EventFrameTriggerFrameID	R	IInteger	R	-	E	Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

Name	Level	Interface	Access	Unit	Visibility	Description
EventExposureEndData	R	ICategory	R	-	E	Category that contains all the data features related to the ExposureEnd Event.
EventExposureEnd	R	IInteger	R	-	E	Returns the unique identifier of the ExposureEnd type of Event.
EventExposureEndTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the ExposureEnd Event.
EventExposureEndFrameID	R	IInteger	R	-	E	Returns the unique Identifier of the Frame (or image) that generated the ExposureEnd Event.

Name	Level	Interface	Access	Unit	Visibility	Description
EventErrorData	R	ICategory	R	-	E	Category that contains all the data features related to the Error Event.
EventError	R	IInteger	R	-	E	Returns the unique identifier of the Error type of Event.
EventErrorTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the Error Event.
EventErrorFrameID	R	IInteger	R	-	E	If applicable, returns the unique Identifier of the Frame (or image) that

						generated the Error Event.
EventErrorCode	R	IInteger	R	-	E	Returns an error code for the error(s) that happened.

2.7 Analog Control

Contains the features related to the video signal conditioning in the analog domain.

Table 2-7: Analog Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
AnalogControl	O	ICategory	R	-	B	Category that contains the Analog control features.
GainSelector	O	IEnumeration	R/W	-	B	Selects which Gain is controlled by the various Gain features.
Gain[GainSelector]	O	IFloat	R/W	-	B	Controls the selected gain as an absolute physical value.
GainAuto[GainSelector]	O	IEnumeration	R/W	-	B	Sets the automatic gain control (AGC) mode.
GainAutoBalance	O	IEnumeration	R/W	-	B	Sets the mode for automatic gain balancing between the sensor color channels or taps.
BlackLevelSelector	O	IEnumeration	R/W	-	E	Selects which Black Level is controlled by the various Black Level features.
BlackLevel[BlackLevelSelector]	O	IFloat	R/W	-	E	Controls the analog black level as an absolute physical value.
BlackLevelAuto[BlackLevelSelector]	O	IEnumeration	R/W	-	E	Controls the mode for automatic black level adjustment.
BlackLevelAutoBalance	O	IEnumeration	R/W	-	E	Controls the mode for automatic black level balancing between the sensor color channels or taps.
WhiteClipSelector	O	IEnumeration	R/W	-	E	Selects which White Clip to control.
WhiteClip[WhiteClipSelector]	O	IFloat	R/W	-	E	Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.
BalanceRatioSelector	O	IEnumeration	R/W	-	E	Selects which Balance ratio to control.
BalanceRatio[BalanceRatioSelector]	O	IFloat	R/W	-	E	Controls ratio of the selected color component to a reference color

						component.
BalanceWhiteAuto	O	IEnumeration	R/W	-	E	Controls the mode for automatic white balancing between the color channels.
Gamma	O	IFloat	R/W	-	B	Controls the gamma correction of pixel intensity.

2.8 LUT Control

Contains the features related to the look-up table (LUT) control.

Table 2-8: Lut Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
LUTControl	O	ICategory	R	-	E	Category that includes the LUT control features.
LUTSelector	O	IEnumeration	R/W	-	E	Selects which LUT to control.
LUTEnable[LUTSelector]	O	IBoolean	R/W	-	E	Activates the selected LUT.
LUTIndex[LUTSelector]	O	IInteger	R/W	-	G	Control the index (offset) of the coefficient to access in the selected LUT.
LUTValue[LUTSelector][LUTIndex]	O	IInteger	R/W	-	G	Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.
LUTValueAll[LUTSelector]	O	IRegister	R/W	-	G	Accesses all the LUT coefficients in a single access without using individual LUTIndex.

2.9 GenICam Control

Contains the features related to GenICam control and access.

Table 2-9: GenICam Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
Root	M	ICategory	R	-	G	Provides the Root of the GenICam features tree.
Device	M	IPort	R/W	-	I	Provides the default GenICam port of the Device.
TLParamsLocked	M	IInteger	R/W	-	I	Used by the Transport Layer to prevent critical features from changing during acquisition.

2.10 Transport Layer Control

Contains the features related to the Transport Layer Control.

Table 2-10: Transport Layer Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
TransportLayerControl	R	ICategory	R	-	B	Category that contains the transport Layer control features.
PayloadSize	R	IInteger	R	B	E	Provides the number of bytes transferred for each image or chunk on the stream channel.
DeviceTapGeometry	R	IEnumeration	R/(W)	-	E	This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

Name	Level	Interface	Access	Unit	Visibility	Description
GigEVision	O	ICategory	R	-	B	Category that contains the features pertaining to the GigE Vision transport layer of the device.
GevVersionMajor	R	IInteger	R	-	E	This feature is deprecated (See DeviceTLVersionMajor).
GevVersionMinor	R	IInteger	R	-	E	This feature is deprecated (See DeviceTLVersionMinor).
GevDeviceModeIsBigEndian	O	IBoolean	R	-	G	This feature is deprecated (See DeviceRegistersEndianess).
GevDeviceClass	O	IEnumeration	R	-	G	This feature is deprecated (See DeviceType).

GevDeviceModeCharacterSet	O	IEnumeration	R	-	G	This feature is deprecated (See DeviceCharacterSet).
GevPhysicalLinkConfiguration	O	IEnumeration	R/W	-	E	Controls the principal physical link configuration to use on next restart/power-up of the device.
GevCurrentPhysicalLinkConfiguration	O	IEnumeration	R	-	E	Indicates the current physical link configuration of the device.
GevActiveLinkCount	O	IInteger	R	-	E	Indicates the current number of active logical links.
GevSupportedOptionSelector	O	IEnumeration	R/W	-	E	Selects the GEV option to interrogate for existing support.
GevSupportedOption[GevSupportedOptionSelector]	O	IBoolean	R	-	E	Returns if the selected GEV option is supported.
GevInterfaceSelector	O	IInteger	R/W	-	B	Selects which logical link to control.
GevLinkSpeed[GevInterfaceSelector]	O	IInteger	R	Mbs	E	This feature is deprecated (See DeviceLinkSpeed).
GevMACAddress[GevInterfaceSelector]	O	IInteger	R	-	B	MAC address of the logical link.
GevPAUSEFrameReception[GevInterfaceSelector]	O	IBoolean	R/(W)	-	E	Controls whether incoming PAUSE Frames are handled on the given logical link.
GevPAUSEFrameTransmission[GevInterfaceSelector]	O	IBoolean	R/(W)	-	E	Controls whether PAUSE Frames can be generated on the given logical link.
GevCurrentIPConfigurationLLA[GevInterfaceSelector]	O	IBoolean	R/W	-	B	Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.
GevCurrentIPConfigurationDHCP[GevInterfaceSelector]	O	IBoolean	R/W	-	B	Controls whether the DHCP IP configuration scheme is activated on the given logical link.
GevCurrentIPConfigurationPersistentIP[GevInterfaceSelector]	O	IBoolean	R/W	-	B	Controls whether the PersistentIP configuration scheme is activated on the given logical link.
GevCurrentIPAddress[GevInterfaceSelector]	O	IInteger	R	-	B	Reports the IP address for the given logical link.
GevCurrentSubnetMask[GevInterfaceSelector]	O	IInteger	R	-	B	Reports the subnet mask of the given logical link.
GevCurrentDefaultGateway[GevInterfaceSelector]	O	IInteger	R	-	B	Reports the default gateway IP address to be used on the given logical link.
GevIPConfigurationStatus[GevInterface	O	IEnumeration	R	-	B	Reports the current IP configuration status.

Selector]						
GevFirstURL	O	IString	R	-	G	Indicates the first URL to the GenICam XML device description file.
GevSecondURL	O	IString	R	-	G	Indicates the second URL to the GenICam XML device description file.
GevNumberOfInterfaces	O	IInteger	R	-	E	This feature is deprecated (See DeviceLinkSelector).
GevPersistentIPAddress[GevInterfaceSelector]	O	IInteger	R/W	-	B	Controls the Persistent IP address for this logical link.
GevPersistentSubnetMask[GevInterfaceSelector]	O	IInteger	R/W	-	B	Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.
GevPersistentDefaultGateway[GevInterfaceSelector]	O	IInteger	R/W	-	B	Controls the persistent default gateway for this logical link.
GevMessageChannelCount	O	IInteger	R	-	E	This feature is deprecated (See DeviceEventChannelCount).
GevStreamChannelCount	O	IInteger	R	-	E	This feature is deprecated (See DeviceStreamChannelCount).
GevHeartbeatTimeout	O	IInteger	R/W	ms	G	This feature is deprecated (See DeviceLinkHeartbeatTimeout).
GevTimestampTickFrequency	O	IInteger	R	Hz	E	This feature is deprecated (See the increment of the TimestampLatchValue feature).
GevTimestampControlLatch	O	ICommand	W	-	E	This feature is deprecated (See TimestampLatch).
GevTimestampControlReset	O	ICommand	W	-	E	This feature is deprecated (See TimestampReset).
GevTimestampValue	O	IInteger	R		E	This feature is deprecated (See TimestampLatchValue).
GevDiscoveryAckDelay	O	IInteger	R/(W)	ms	E	Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.
GevIEEE1588	O	IBoolean	R/W	-	E	Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.
GevIEEE1588ClockAccuracy	O	IEnumeration	R/(W)	-	E	Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.
GevIEEE1588Status	O	IEnumeration	R	-	E	Provides the status of the IEEE 1588 clock.
GevGVCPExtendedStatusCodesSelector	O	IEnumeration	R/W	-	G	Selects the GigE Vision version to control extended status codes for.
GevGVCPExtendedStatusCodes[GevG	O	IBoolean	R/W	-	G	Enables the generation of extended status codes.

VCPExtendedStatusCodesSelector]						
GevGVCPPendingAck	O	IBoolean	R/W	-	G	Enables the generation of PENDING_ACK.
GevGVCPHeartbeatDisable	O	IBoolean	R/W	-	E	This feature is deprecated (See DeviceHeartbeatMode).
GevGVCPPendingTimeout	O	IInteger	R	-	G	This feature is deprecated (See DeviceCommandTimeout).
GevPrimaryApplicationSwitchoverKey	O	IInteger	W-O	-	G	Controls the key to use to authenticate primary application switchover requests.
GevGVSPExtendedIDMode	O	IEnumeration	R/(W)	-	E	Enables the extended IDs mode.
GevCCP	O	IEnumeration	R/W	-	G	Controls the device access privilege of an application.
GevPrimaryApplicationSocket	O	IInteger	R	-	G	Returns the UDP source port of the primary application.
GevPrimaryApplicationIPAddress	O	IInteger	R	-	G	Returns the address of the primary application.
GevMCPHostPort	O	IInteger	R/W	-	G	Controls the port to which the device must send messages.
GevMCDA	O	IInteger	R/W	-	G	Controls the destination IP address for the message channel.
GevMCTT	O	IInteger	R/W	ms	G	Provides the transmission timeout value in milliseconds.
GevMCRC	O	IInteger	R/W	-	G	Controls the number of retransmissions allowed when a message channel message times out.
GevMCSP	O	IInteger	R	-	G	This feature indicates the source port for the message channel.
GevStreamChannelSelector	O	IInteger	R/W	-	E	Selects the stream channel to control.
GevSCCFGPacketResendDestination[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Enables the alternate IP destination for stream packets resent due to a packet resend request.
GevSCCFGAllInTransmission[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.
GevSCCFGUnconditionalStreaming[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).
GevSCCFGExtendedChunkData[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Enables cameras to use the extended chunk data payload type for this stream channel.
GevSCPDirection[GevStreamChannelSe	O	IEnumeration	R	-	G	This feature is deprecated (See DeviceStreamChannelType).

lector]						
GevSCPInterfaceIndex[GevStreamChannelSelector]	O	IInteger	R/(W)	-	G	Index of the logical link to use.
GevSCPHostPort[GevStreamChannelSelector]	O	IInteger	R/W	-	G	Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.
GevSCPSFireTestPacket[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Sends a test packet.
GevSCPSDoNotFragment[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	The state of this feature is copied into the "do not fragment" bit of IP header of each stream packet.
GevSCPSBigEndian[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	This feature is deprecated (See DeviceStreamChannelEndianness).
GevSCPSPacketSize[GevStreamChannelSelector]	R	IInteger	R/(W)	B	E	This feature is deprecated (See DeviceStreamChannelPacketSize).
GevSCPD[GevStreamChannelSelector]	R	IInteger	R/W		E	Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.
GevSCDA[GevStreamChannelSelector]	O	IInteger	R/W	-	G	Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.
GevSCSP[GevStreamChannelSelector]	O	IInteger	R	-	G	Indicates the source port of the stream channel.
GevSCZoneCount[GevStreamChannelSelector]	O	IInteger	R	-	G	Reports the number of zones per block transmitted on the selected stream channel.
GevSCZoneDirectionAll[GevStreamChannelSelector]	O	IInteger	R	-	G	Reports the transmission direction of each zone transmitted on the selected stream channel.
GevSCZoneConfigurationLock[GevStreamChannelSelector]	O	IBoolean	R/W	-	G	Controls whether the selected stream channel multi-zone configuration is locked.

Name	Level	Interface	Access	Unit	Visibility	Description

NetworkStatistics	O	ICategory	R	-	G	Category that contains statistics pertaining to various modules of the GigE Vision transport layer.

Name	Level	Interface	Access	Unit	Visibility	Description
oMACControlFunctionEntity	O	ICategory	R	-	G	Category that contains statistics pertaining to the MAC control PAUSE function of the device.
aPAUSEMACCtrlFramesTransmitted[GevInterfaceSelector]	O	IInteger	R	-	G	Reports the number of transmitted PAUSE frames.
aPAUSEMACCtrlFramesReceived[GevInterfaceSelector]	O	IInteger	R	-	G	Reports the number of received PAUSE frames.

Name	Level	Interface	Access	Unit	Visibility	Description
CameraLink	O	ICategory	R	-	B	Category that contains the features pertaining to the Camera Link transport layer of the device.
ClConfiguration	R	IEnumeration	R/(W)	-	B	This Camera Link specific feature describes the configuration used by the camera.
ClTimeSlotsCount	O	IEnumeration	R/(W)	-	E	This Camera Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Name	Level	Interface	Access	Unit	Visibility	Description
CoaXPress	O	ICategory	R	-	B	Category that contains the features pertaining to the CoaXPress transport layer of the device.
CxpLinkConfigurationStatus	R	IEnumeration	R	-	B	This feature indicates the current and active Link configuration used by the Device.
CxpLinkConfigurationPreferred	R	IEnumeration	R	-	E	Provides the Link configuration that allows the Transmitter Device to

						operate in its default mode.
CxpLinkConfiguration	R	IEnumeration	R/W	-	B	This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.
CxpConnectionSelector	R	IInteger	R/W	-	E	Selects the CoaXPress physical connection to control.
CxpConnectionTestMode[CxpConnectionSelector]	R	IEnumeration	R/W	-	E	Enables the test mode for an individual physical connection of the Device.
CxpConnectionTestErrorCount[CxpConnectionSelector]	R	IInteger	R/W	-	E	Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.
CxpConnectionTestPacketCount[CxpConnectionSelector]	R	IInteger	R/W	-	E	Reports the current count for test packets recieved by the device on the connection selected by CxpConnectionSelector.
CxpPoCxpAuto	O	ICommand	W	-	E	Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.
CxpPoCxpTurnOff	O	ICommand	W	-	E	Disable Power over CoaXPress (PoCXP) for the Link.
CxpPoCxpTripReset	O	ICommand	W	-	E	Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).
CxpPoCxpStatus	O	IEnumeration	R	-	E	Returns the Power over CoaXPress (PoCXP) status of the Device.

2.11 User Set Control

Contains the features related to the User Set Control to save and load the user device settings.

Table 2-11: User Set Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
UserSetControl	R	ICategory	R	-	B	Category that contains the User Set control features.
UserSetSelector	R	IEnumeration	R/W	-	B	Selects the feature User Set to load, save or configure.
UserSetLoad[UserSetSelector]	R	ICommand	(R)/W	-	B	Loads the User Set specified by UserSetSelector to the device and makes it active.

UserSetSave[UserSetSelector]	R	ICommand	(R)/W	-	B	Save the User Set specified by UserSetSelector to the non-volatile memory of the device.
UserSetDefault	O	IEnumeration	R/W	-	B	Selects the feature User Set to load and make active by default when the device is reset.
UserSetDefaultSelector	O	IEnumeration	R/W	-	B	This feature is deprecated (See UserSetDefault).
UserSetFeatureSelector	R	IEnumeration	R/W	-	E	Selects which individual UserSet feature to control.
UserSetFeatureEnable[UserSetFeatureSelector]	R	IBoolean	R/(W)	-	E	Enables the selected feature and make it active in all the UserSets.

2.12 Chunk Data Control

Contains the features related to the Chunk Data Control.

Table 2-12: Chunk Data Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
ChunkDataControl	R	ICategory	R	-	E	Category that contains the Chunk Data control features.
ChunkModeActive	R	IBoolean	R/W	-	E	Activates the inclusion of Chunk data in the payload of the image.
ChunkSelector	R	IEnumeration	R/W	-	E	Selects which Chunk to enable or control.
ChunkEnable[ChunkSelector]	R	IBoolean	R/W	-	E	Enables the inclusion of the selected Chunk data in the payload of the image.
ChunkImage	R	IRegister	R	-	G	Returns the entire image data included in the payload.
ChunkOffsetX	R	IInteger	R	-	E	Returns the OffsetX of the image included in the payload.
ChunkOffsetY	R	IInteger	R	-	E	Returns the OffsetY of the image included in the payload.
ChunkWidth	R	IInteger	R	-	E	Returns the Width of the image included in the payload.
ChunkHeight	R	IInteger	R	-	E	Returns the Height of the image included in the payload.
ChunkPixelFormat	R	IEnumeration	R	-	E	Returns the PixelFormat of the image included in the payload.

ChunkPixelDynamicRangeMin	R	IInteger	R	-	E	Returns the minimum value of dynamic range of the image included in the payload.
ChunkPixelDynamicRangeMax	R	IInteger	R	-	E	Returns the maximum value of dynamic range of the image included in the payload.
ChunkTimestamp	R	IInteger	R	-	E	Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event.
ChunkTimestampLatchValue	R	IInteger	R	ns	E	Returns the last Timestamp latched with the TimestampLatch command.
ChunkLineStatusAll	R	IInteger	R	-	E	Returns the status of all the I/O lines at the time of the FrameStart internal event.
ChunkCounterSelector	R	IEnumeration	R/W	-	E	Selects which counter to retrieve data from.
ChunkCounterValue[ChunkCounterSelector]	R	IInteger	R	-	E	Returns the value of the selected Chunk counter at the time of the FrameStart event.
ChunkTimerSelector	R	IEnumeration	R/W	-	E	Selects which Timer to retrieve data from.
ChunkTimerValue[ChunkTimerSelector]	R	IFloat	R	us	E	Returns the value of the selected Timer at the time of the FrameStart internal event.
ChunkExposureTime	R	IFloat	R	us	E	Returns the exposure time used to capture the image.
ChunkGainSelector	R	IEnumeration	R/W	-	E	Selects which Gain to return.
ChunkGain[ChunkGainSelector]	R	IFloat	R	-	E	Returns the gain used to capture the image.
ChunkBlackLevelSelector	R	IEnumeration	R/W	-	E	Selects which Black Level to return.
ChunkBlackLevel[ChunkBlackLevelSelector]	R	IFloat	R	-	E	Returns the black level used to capture the image included in the payload.
ChunkLinePitch	R	IInteger	R	B	E	Returns the LinePitch of the image included in the payload.
ChunkFrameID	R	IInteger	R	-	E	Returns the unique Identifier of the frame (or image) included in the payload.
ChunkSourceID	O	IEnumeration	R	-	E	Returns the identifier of Source that the image comes from.
ChunkRegionID	O	IEnumeration	R	-	E	Returns the identifier of Region that the image comes from.
ChunkTransferBlockID	R	IInteger	R	-	E	Returns the unique identifier of the transfer block used to transport the

						payload.
ChunkTransferStreamID	R	IEnumeration	R	-	E	Returns identifier of the stream that generated this block.
ChunkTransferQueueCurrentBlockCount	O	IInteger	R	-	E	Returns the current number of blocks in the transfer queue.
ChunkStreamChannelID	R	IInteger	R	-	E	Returns identifier of the stream channel used to carry the block.
ChunkSequencerSetActive	R	IInteger	R	-	E	Return the index of the active set of the running sequencer included in the payload.

2.13 File Access Control

Contains the features related to the File that provides all the sevices necessary for generic file access of a device.

Table 2-13: File Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
FileAccessControl	R	ICategory	R	-	G	Category that contains the File Access control features.
FileSelector	R	IEnumeration	R/(W)	-	G	Selects the target file in the device.
FileOperationSelector[FileSelector]	R	IEnumeration	R/W	-	G	Selects the target operation for the selected file in the device.
FileOperationExecute[FileSelector][FileOperationSelector]	R	ICommand	(R)/W	-	G	Executes the operation selected by FileOperationSelector on the selected file.
FileOpenMode[FileSelector]	R	IEnumeration	R/(W)	-	G	Selects the access mode in which a file is opened in the device.
FileAccessBuffer	R	IRegister	R/(W)	-	G	Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
FileAccessOffset[FileSelector][FileOperationSelector]	R	IInteger	R/(W)	B	G	Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.
FileAccessLength[FileSelector][FileOperationSelector]	R	IInteger	R/W	B	G	Controls the Length of the mapping between the device file storage and the FileAccessBuffer.
FileOperationStatus[FileSelector][FileO	R	IEnumeration	R	-	G	Represents the file operation execution status.

perationSelector]						
FileOperationResult[FileSelector][FileOperationSelector]	R	IInteger	R	-	G	Represents the file operation result.
FileSize[FileSelector]	R	IInteger	R	B	G	Represents the size of the selected file in bytes.

2.14 Color Transformation Control

Contains the features related to the control of the color transformation.

Table 2-14: Color Transformation summary

Name	Level	Interface	Access	Unit	Visibility	Description
ColorTransformationControl	R	ICategory	R	-	E	Category that contains the Color Transformation control features.
ColorTransformationSelector	O	IEnumeration	R/W	-	E	Selects which Color Transformation module is controlled by the various Color Transformation features.
ColorTransformationEnable[ColorTransformationSelector]	O	IBoolean	R/W	-	E	Activates the selected Color Transformation module.
ColorTransformationValueSelector[ColorTransformationSelector]	O	IEnumeration	R/W	-	E	Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module.
ColorTransformationValue[ColorTransformationSelector][ColorTransformationValueSelector]	O	IFloat	R/W	-	E	Represents the value of the selected Gain factor or Offset inside the Transformation matrix.

2.15 Action Control

Contains the features related to the control of the Action command mechanism.

Table 2-15: Action Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
ActionControl	R	ICategory	R	-	G	Category that contains the Action control features.
ActionUnconditionalMode	O	IEnumeration	R/W	-	G	Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.
ActionDeviceKey	O	IInteger	W-O	-	G	Provides the device key that allows the device to check the validity of action commands.
ActionQueueSize	O	IInteger	R	-	G	Indicates the size of the scheduled action commands queue.
ActionSelector	O	IInteger	R/W	-	G	Selects to which Action Signal further Action settings apply.
ActionGroupMask[ActionSelector]	O	IInteger	R/W	-	G	Provides the mask that the device will use to validate the action on reception of the action protocol message.
ActionGroupKey[ActionSelector]	O	IInteger	R/W	-	G	Provides the key that the device will use to validate the action on reception of the action protocol message.

2.16 Source Control

Contains the features related to the control of the multiple Source devices.

Table 2-16: Source Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
SourceControl	O	ICategory	R	-	E	Category that contains the source control features.
SourceCount	O	IInteger	R/(W)	-	E	Controls or returns the number of sources supported by the device.
SourceSelector	O	IEnumeration	R/W	-	E	Selects the source to control.

2.17 Transfer Control

Contains the features related to the control of the Transfers.

Table 2-17: Transfer Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
TransferControl	R	ICategory	R	-	E	Category for the data Transfer Control features.
TransferSelector	O	IEnumeration	R/(W)	-	E	Selects which stream transfers are currently controlled by the selected Transfer features.
TransferControlMode[TransferSelector]	R	IEnumeration	R/(W)	-	E	Selects the control method for the transfers.
TransferOperationMode[TransferSelector]	O	IEnumeration	R/(W)	-	E	Selects the operation mode of the transfer.
TransferBlockCount[TransferSelector]	O	IInteger	R/(W)	-	E	Specifies the number of data Blocks that the device should stream before stopping.
TransferBurstCount	O	IInteger	R/W	-	E	Number of Block(s) to transfer for each TransferBurstStart trigger.
TransferQueueMaxBlockCount[TransferSelector]	O	IInteger	R/(W)	-	E	Controls the maximum number of data blocks that can be stored in the block queue of the selected stream.
TransferQueueCurrentBlockCount[TransferSelector]	O	IInteger	R	-	E	Returns the number of Block(s) currently in the transfer queue.
TransferQueueMode[TransferSelector]	O	IEnumeration	R/(W)	-	E	Specifies the operation mode of the transfer queue.
TransferStart[TransferSelector]	O	ICommand	(R)/W	-	E	Starts the streaming of data blocks out of the device.
TransferStop[TransferSelector]	O	ICommand	(R)/W	-	E	Stops the streaming of data Block(s).
TransferAbort[TransferSelector]	O	ICommand	(R)/W	-	E	Aborts immediately the streaming of data block(s).
TransferPause[TransferSelector]	O	ICommand	(R)/W	-	G	Pauses the streaming of data Block(s).
TransferResume[TransferSelector]	O	ICommand	(R)/W	-	G	Resumes a data Blocks streaming that was previously paused by a TransferPause command.
TransferTriggerSelector[TransferSelector]	O	IEnumeration	R/W	-	G	Selects the type of transfer trigger to configure.
TransferTriggerMode[TransferSelector][	R	IEnumeration	R/W	-	G	Controls if the selected trigger is active.

TransferTriggerSelector]						
TransferTriggerSource[TransferTriggerSelector]	O	IEnumeration	R/W	-	G	Specifies the signal to use as the trigger source for transfers.
TransferTriggerActivation [TransferTriggerSelector]	O	IEnumeration	R/W	-	G	Specifies the activation mode of the transfer control trigger.
TransferStatusSelector[TransferSelector]	R	IEnumeration	R/W	-	G	Selects which status of the transfer module to read.
TransferStatus[TransferStatusSelector]	R	IBool	R	-	G	Reads the status of the Transfer module signal selected by TransferStatusSelector.
TransferComponentSelector[TransferSel ector]	O	IEnumeration	R/W	-	G	Selects the color component for the control of the TransferStreamChannel feature.
TransferStreamChannel[TransferSelecto r][TransferComponentSelector]	O	IInteger	R/W	-	G	Selects the streaming channel that will be used to transfer the selected stream of data.

2.18 Sequencer Control

Contains the features related to the control of the Sequencer for features change.

Table 2-18: Sequencer Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
SequencerControl	O	ICategory	R	-	E	Category for the Sequencer Control features.
SequencerMode	R	IEnumeration	R/W	-	E	Controls if the sequencer mechanism is active.
SequencerConfigurationMode	R	IEnumeration	R/W	-	E	Controls if the sequencer configuration mode is active.
SequencerFeatureSelector	R	IEnumeration	R/W	-	E	Selects which sequencer features to control.
SequencerFeatureEnable[SequencerFeat ureSelector]	R	IBoolean	R/(W)	-	E	Enables the selected feature and make it active in all the sequencer sets.
SequencerSetSelector	R	IInteger	R/W	-	E	Selects the sequencer set to which further feature settings applies.

SequencerSetSave[SequencerSetSelector]	R	ICommand	(R)/W	-	E	Saves the current device state to the sequencer set selected by the SequencerSetSelector.
SequencerSetLoad[SequencerSetSelector]	R	ICommand	(R)/W	-	E	Loads the sequencer set selected by SequencerSetSelector in the device.
SequencerSetActive	O	IInteger	R	-	E	Contains the currently active sequencer set.
SequencerSetStart	R	IInteger	R/W	-	E	Sets the initial/start sequencer set, which is the first set used within a sequencer.
SequencerPathSelector[SequencerSetSelector]	R	IInteger	R/W	-	E	Selects to which branching path further path settings applies.
SequencerSetNext[SequencerSetSelector] [SequencerPathSelector]	R	IInteger	R/W	-	E	Specifies the next sequencer set.
SequencerTriggerSource[SequencerSetSelector] [SequencerPathSelector]	R	IEnumeration	R/W	-	E	Specifies the internal signal or physical input line to use as the sequencer trigger source.
SequencerTriggerActivation[SequencerSetSelector] [SequencerPathSelector]	R	IEnumeration	R/W	-	E	Specifies the activation mode of the sequencer trigger.

2.19 Software Signal Control

Contains the features related to the control of the Software Signal.

Table 2-19: Software Signal Control Summary

Name	Level	Interface	Access	Unit	Visibility	Description
SoftwareSignalControl	O	ICategory	R	-	B	Category that contains the Software Signal Control features.
SoftwareSignalSelector	O	IEnumeration	R/W	-	B	Selects which Software Signal features to control.
SoftwareSignalPulse[SoftwareSignalSelector]	O	ICommand	(R)/W	-	B	Generates a pulse signal that can be used as a software trigger.

3 Device Control

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

3.1 DeviceControl

Name	DeviceControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category for device information and control.

3.2 DeviceType

Name	DeviceType
Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Guru
Values	Transmitter Receiver Transceiver Peripheral

Returns the device type.

Possible values are:

- **Transmitter**: Data stream transmitter device.
- **Receiver**: Data stream receiver device.
- **Transceiver**: Data stream receiver and transmitter device.
- **Peripheral**: Controlable device (with no data stream handling).

3.3 DeviceScanType

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

Scan type of the sensor of the device.

Typically, this feature is not writable. But some cameras might allow switching between linescan and areascan.

Possible values are:

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

3.4 DeviceVendorName

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

Name of the manufacturer of the device.

3.5 DeviceModelName

Name	DeviceModelName
Category	DeviceControl
Level	Recommended
Interface	IString

Access	Read
Unit	-
Visibility	Beginner
Values	Any NULL-terminated string

Model of the device.

3.6 DeviceFamilyName

Name	DeviceFamilyName
Category	DeviceControl
Level	Optional
Interface	IString
Access	Read
Unit	-
Visibility	Beginner
Values	Any NULL-terminated string

Identifier of the product family of the device.

3.7 DeviceManufacturerInfo

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

Manufacturer information about the device.

3.8 DeviceVersion

Name	DeviceVersion
Category	DeviceControl
Level	Recommended
Interface	IString

Access	Read
Unit	-
Visibility	Beginner
Values	Any NULL-terminated string

Version of the device.

3.9 DeviceFirmwareVersion

Name	DeviceFirmwareVersion
Category	DeviceControl
Level	Recommended
Interface	IString
Access	Read
Unit	-
Visibility	Beginner
Values	Any NULL-terminated string

Version of the firmware in the device.

3.1 DeviceSerialNumber

Name	DeviceSerialNumber
Category	DeviceControl
Level	Recommended
Interface	IString
Access	Read
Unit	-
Visibility	Expert
Values	Any NULL-terminated string

Device`s serial number. This string is a unique identifier of the device.

3.2DeviceID (Deprecated)

Name	DeviceID
Category	DeviceControl
Level	Recommended
Interface	IString

Access	Read
Unit	-
Visibility	Invisible
Values	Any NULL-terminated string

This feature is deprecated (See DeviceSerialNumber). It was representing the Device unique identifier (serial number).

To help backward compatibility, this feature can be included as Invisible in the device`s XML.

3.3 DeviceUserID

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

User-programmable device identifier.

When this feature is present, it must be writable. The recommended factory default value is an empty string.

3.4 DeviceSFNCVersionMajor

Name	DeviceSFNCVersionMajor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	>0

Major version of the Standard Features Naming Convention that was used to create the device`s GenICam XML.

3.5 DeviceSFNCVersionMinor

Name	DeviceSFNCVersionMinor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	≥0

Minor version of the Standard Features Naming Convention that was used to create the device`s GenICam XML.

3.6 DeviceSFNCVersionSubMinor

Name	DeviceSFNCVersionSubMinor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	≥0

Sub minor version of Standard Features Naming Convention that was used to create the device`s GenICam XML.

3.7 DeviceManifestEntrySelector

Name	DeviceManifestEntrySelector
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru

Values	≥1

Selects the manifest entry to reference.

3.8 DeviceManifestXMLMajorVersion

Name	DeviceManifestXMLMajorVersion[DeviceManifestEntrySelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Indicates the major version number of the GenICam XML file of the selected manifest entry.

3.9 DeviceManifestXMLMinorVersion

Name	DeviceManifestXMLMinorVersion[DeviceManifestEntrySelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Indicates the minor version number of the GenICam XML file of the selected manifest entry.

3.10 DeviceManifestXMLSubMinorVersion

Name	DeviceManifestXMLSubMinorVersion[DeviceManifestEntrySelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru

Values	≥0

Indicates the subminor version number of the GenICam XML file of the selected manifest entry.

3.11 DeviceManifestSchemaMajorVersion

Name	DeviceManifestSchemaMajorVersion[DeviceManifestEntrySelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Indicates the major version number of the schema file of the selected manifest entry.

3.12 DeviceManifestSchemaMinorVersion

Name	DeviceManifestSchemaMinorVersion[DeviceManifestEntrySelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Indicates the minor version number of the schema file of the selected manifest entry.

3.13 DeviceManifestPrimaryURL

Name	DeviceManifestPrimaryURL[DeviceManifestEntrySelector]
Category	DeviceControl
Level	Optional
Interface	IString
Access	Read
Unit	-
Visibility	Guru

Values	-

Indicates the first URL to the GenICam XML device description file of the selected manifest entry.

3.14 DeviceManifestSecondaryURL

Name	DeviceManifestSecondaryURL[DeviceManifestEntrySelector]
Category	DeviceControl
Level	Optional
Interface	IString
Access	Read
Unit	-
Visibility	Guru
Values	-

Indicates the second URL to the GenICam XML device description file of the selected manifest entry.

3.15 DeviceTLType

Name	DeviceTLType
Category	DeviceControl
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Beginner
Values	GigEVision CameraLink CameraLinkHS CoaXPress USB3Vision Custom

Transport Layer type of the device.

Possible values are:

- **GigEVision**: GigE Vision.
- **CameraLink:** Camera Link.
- **CameraLinkHS:** Camera Link High Speed.

- **CoaXPress:** CoaXPress.
- **USB3Vision:** USB3 Vision.
- **Custom:** Custom Transport Layer.

3.16 DeviceTLVersionMajor

Name	DeviceTLVersionMajor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	>0

Major version of the Transport Layer of the device.

3.17 DeviceTLVersionMinor

Name	DeviceTLVersionMinor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	≥0

Minor version of the Transport Layer of the device.

3.18 DeviceTLVersionSubMinor

Name	DeviceTLVersionSubMinor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read

Unit	-
Visibility	Beginner
Values	≥0

Sub minor version of the Transport Layer of the device.

3.19 DeviceGenCPVersionMajor

Name	DeviceGenCPVersionMajor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	>0

Major version of the GenCP protocol supported by the device.

3.20 DeviceGenCPVersionMinor

Name	DeviceGenCPVersionMinor
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	≥0

Minor version of the GenCP protocol supported by the device.

3.21 DeviceMaxThroughput

Name	DeviceMaxThroughput
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read

Unit	Bps
Visibility	Expert
Values	>0

Maximum bandwidth of the data that can be streamed out of the device. This can be used to estimate if the physical connection(s) can sustain transfer of free-running images from the camera at its maximum speed.

3.22 DeviceConnectionSelector

Name	DeviceConnectionSelector
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	≥0

Selects which Connection of the device to control.

3.23 DeviceConnectionSpeed

Name	DeviceConnectionSpeed[DeviceConnectionSelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	Bps
Visibility	Expert
Values	>0

Indicates the speed of transmission of the specified Connection

3.24 DeviceConnectionStatus

Name	DeviceConnectionStatus[DeviceConnectionSelector]
Category	DeviceControl
Level	Optional
Interface	IEnumeration

Access	Read
Unit	-
Visibility	Expert
Values	Active Inactive

Indicates the status of the specified Connection.

Possible values are:

- **Active**: Connection is in use.
- **Inactive**: Connection is not in use.

3.25 DeviceLinkSelector

Name	DeviceLinkSelector
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	≥0

Selects which Link of the device to control.

Generally, a device has only one Link that can be composed of one or many connections. But if there are many, this selector can be used to target a particular Link of the device with certain features.

3.26 DeviceLinkSpeed

Name	DeviceLinkSpeed[DeviceLinkSelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	Bps
Visibility	Expert
Values	≥0

Indicates the speed of transmission negotiated on the specified Link.

Note that this represents the total speed of all the connections of the Link.

3.27 DeviceLinkThroughputLimitMode

Name	DeviceLinkThroughputLimitMode[DeviceLinkSelector]
Category	DeviceControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	On Off

Controls if the **DeviceLinkThroughputLimit** is active. When disabled, lower level TL specific features are expected to control the throughput. When enabled, **DeviceLinkThroughputLimit** controls the overall throughput.

Possible values are:

- **On:** Enables the **DeviceLinkThroughputLimit** feature**.**
- **Off:** Disables the **DeviceLinkThroughputLimit** feature**.**

3.28 DeviceLinkThroughputLimit

Name	DeviceLinkThroughputLimit[DeviceLinkSelector]
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	Bps
Visibility	Expert
Values	≥ 0

Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link. If necessary, delays will be uniformly inserted between transport layer packets in order to control the peak bandwidth.

If the device uses many connections to transmit the data, the feature represents the sum of all the traffic and the bandwidth should be distributed uniformly on the various connections.

Any Transport Layer specific bandwidth controls should be kept in sync with this control as much as possible.

3.29 DeviceLinkConnectionCount

Name	DeviceLinkConnectionCount[DeviceLinkSelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	≥0

Returns the number of physical connection of the device used by a particular Link.

3.30 DeviceLinkHeartbeatMode

Name	DeviceLinkHeartbeatMode[DeviceLinkSelector]
Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	On Off

Activate or deactivate the Link`s heartbeat.

Possible values are:

- **Off:** Disables the Link heartbeat**.**
- **On:** Enables the Link heartbeat**.**

3.31 DeviceLinkHeartbeatTimeout

Name	DeviceLinkHeartbeatTimeout[DeviceLinkSelector]
Category	DeviceControl
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	us

Visibility	Guru
Values	>0

Controls the current heartbeat timeout of the specific Link.

3.32 DeviceLinkCommandTimeout

Name	DeviceCommandTimeout [DeviceLinkSelector]
Category	DeviceControl
Level	Optional
Interface	IFloat
Access	Read
Unit	us
Visibility	Guru
Values	>0

Indicates the command timeout of the specified Link. This corresponds to the maximum response time of the device for a command sent on that link.

3.33 DeviceStreamChannelCount

Name	DeviceStreamChannelCount
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Indicates the number of streaming channels supported by the device.

3.34 DeviceStreamChannelSelector

Name	DeviceStreamChannelSelector
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read/Write

Unit	-
Visibility	Expert
Values	≥0

Selects the stream channel to control.

3.35 DeviceStreamChannelType

Name	DeviceStreamChannelType[DeviceStreamChannelSelector]
Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Guru
Values	Transmitter Receiver

Reports the type of the stream channel.

Possible values are:

- **Transmitter**: Data stream transmitter channel.
- **Receiver**: Data stream receiver channel.

3.36 DeviceStreamChannelLink

Name	DeviceStreamChannelLink[DeviceStreamChannelSelector]
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Guru
Values	≥0

Index of device`s Link to use for streaming the specifed stream channel.

3.37 DeviceStreamChannelEndianness

Name	DeviceStreamChannelEndianness [DeviceStreamChannelSelector]

Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Guru
Values	Big Little

Endianess of multi-byte pixel data for this stream.

Possible values are:

- **Big**: Stream channel data is big Endian.
- **Little**: Stream channel data is little Endian.

3.38 DeviceStreamChannelPacketSize

Name	DeviceStreamChannelPacketSize [DeviceStreamChannelSelector]
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	B
Visibility	Expert
Values	>0

Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.

3.39 DeviceEventChannelCount

Name	DeviceEventChannelCount
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Indicates the number of event channels supported by the device.

3.40 DeviceMessageChannelCount (Deprecated)

Name	DeviceMessageChannelCount
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

This feature is deprecated (See DeviceEventChannelCount). It indicates the number of message/event channels supported by the device.

3.41 DeviceCharacterSet

Name	DeviceCharacterSet
Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Guru
Values	UTF8 ASCII

Character set used by the strings of the device`s bootstrap registers.

Possible values are:

- **UTF8**: Device use UTF8 character set.
- **ASCII**: Device use ASCII character set.

3.42 DeviceReset

Name	DeviceReset
Category	DeviceControl
Level	Recommended
Interface	ICommand

Access	Write
Unit	-
Visibility	Guru
Values	-

Resets the device to its power up state. After reset, the device must be rediscovered.

Note that some Transport Layers require the acknowledgement of the DeviceReset command before starting the actual reset of the device.

3.43 DeviceIndicatorMode

Name	DeviceIndicatorMode
Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Inactive Active ErrorStatus

Controls the behavior of the indicators (such as LEDs) showing the status of the Device.

Possible values are:

- **Inactive**: Device`s indicators are inactive (Off).
- **Active**: Device`s indicators are active showing their respective status.
- **ErrorStatus:** Device`s indicators are inactive unless an error occurs.

3.44 DeviceFeaturePersistenceStart

Name	DeviceFeaturePersistenceStart
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Guru
Values	-

Indicate to the device and GenICam XML to get ready for persisting of all streamable features.

Note that device persistence is done by reading the device features and saving them outside of the device.

3.45 DeviceFeaturePersistenceEnd

Name	DeviceFeaturePersistenceEnd
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Guru
Values	-

Indicate to the device the end of feature persistence.

3.46 DeviceRegistersStreamingStart

Name	DeviceRegistersStreamingStart
Category	DeviceControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Guru
Values	-

Prepare the device for registers streaming without checking for consistency.

If the camera implements this feature, GenApi guarantees using it to announce register streaming.

If the feature is present, but currently not writable (locked), the application must not start register streaming and must avoid switching the access mode and range verification off until the feature becomes writable again.

3.47 DeviceRegistersStreamingEnd

Name	DeviceRegistersStreamingEnd
Category	DeviceControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write

Unit	-
Visibility	Guru
Values	-

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.

3.48 DeviceRegistersCheck

Name	DeviceRegistersCheck
Category	DeviceControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Perform the validation of the current register set for consistency. This will update the **DeviceRegistersValid** flag.

3.49 DeviceRegistersValid

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

Returns if the current register set is valid and consistent.

3.50 DeviceRegistersEndianness

Name	DeviceRegistersEndianness
Category	DeviceControl
Level	Optional

Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Guru
Values	Big Little

Endianess of the registers of the device.

Possible values are:

- **Big**: Device's registers are big Endian.
- **Little**: Device's registers are little Endian.

3.51 DeviceTemperatureSelector

Name	DeviceTemperatureSelector
Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Sensor Mainboard Device-specific

Selects the location within the device, where the temperature will be measured.

Possible values are:

- Sensor: Temperature of the image sensor of the camera.
- Mainboard: Temperature of the device`s mainboard.

3.52 DeviceTemperature

Name	DeviceTemperature[DeviceTemperatureSelector]
Category	DeviceControl
Level	Optional
Interface	IFloat
Access	Read
Unit	C

Visibility	Expert
Values	Device-specific

Device temperature in degrees Celsius (C). It is measured at the location selected by DeviceTemperatureSelector.

3.53 DeviceClockSelector

Name	DeviceClockSelector
Category	DeviceControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Sensor SensorDigitization CameraLink Device-specific

Selects the clock frequency to access from the device.

Possible values are:

- **Sensor:** Clock frequency of the image sensor of the camera.
- **SensorDigitization**: Clock frequency of the camera A/D conversion stage.
- **CameraLink**: Frequency of the Camera Link clock.

3.54 DeviceClockFrequency

Name	DeviceClockFrequency[DeviceClockSelector]
Category	DeviceControl
Level	Optional
Interface	IFloat
Access	Read/(Write)
Unit	Hz
Visibility	Expert
Values	≥ 0

Returns the frequency of the selected Clock.

3.55 DeviceSerialPortSelector

Name	DeviceSerialPortSelector
Category	DeviceControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	CameraLink Device-specific

Selects which serial port of the device to control.

Possible values are:

- CameraLink: Serial port associated to the Camera link connection.

3.56 DeviceSerialPortBaudRate

Name	DeviceSerialPortBaudRate[DeviceSerialPortSelector]
Category	DeviceControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Baud9600 Baud19200 Baud38400 Baud57600 Baud115200 Baud230400 Baud460800 Baud921600 …

This feature controls the baud rate used by the selected serial port.

Typical values listed should be used whenever possible. Arbitrary values can also be used by defining new enumeration entries.

Possible values are:

- Baud9600: Serial port speed of 9600 baud.
- Baud19200: Serial port speed of 19200 baud.
- Baud38400: Serial port speed of 38400 baud.
- Baud57600: Serial port speed of 57600 baud.
- Baud115200: Serial port speed of 115200 baud.
- Baud230400: Serial port speed of 230400 baud.
- Baud460800: Serial port speed of 460800 baud.
- Baud921600: Serial port speed of 921600 baud.

3.57 Timestamp

Name	Timestamp
Category	DeviceControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	ns
Visibility	Expert
Values	≥0

Reports the current value of the device timestamp counter.

The same timestamp counter is used for tagging images, chunk and event data.

Note that the increment of the Timestamp feature must correspond to the resolution of the devices's timestamp in nanoseconds.

3.58 TimestampReset

Name	TimestampReset
Category	DeviceControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Resets the current value of the device timestamp counter.

After executing this command, the timestamp counter restarts automatically.

3.59 TimestampLatch

Name	TimestampLatch
Category	DeviceControl
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Visibility	Expert
Values	-

Latches the current timestamp counter into TimestampLatchValue.

3.60 TimestampLatchValue

Name	TimestampLatchValue
Category	DeviceControl
Level	Optional
Interface	IInteger
Access	Read
Unit	ns
Visibility	Expert
Values	≥0

Returns the latched value of the timestamp counter.

Note that the increment of the TimestampLatchValue feature must correspond to the resolution of the devices's timestamp in nanoseconds.

4 Image Format Control

This chapter 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 device has a Source of data that generates a single rectangular image. This image can be entirely or partially streamed out of the device using one or many Region of interest (ROI).

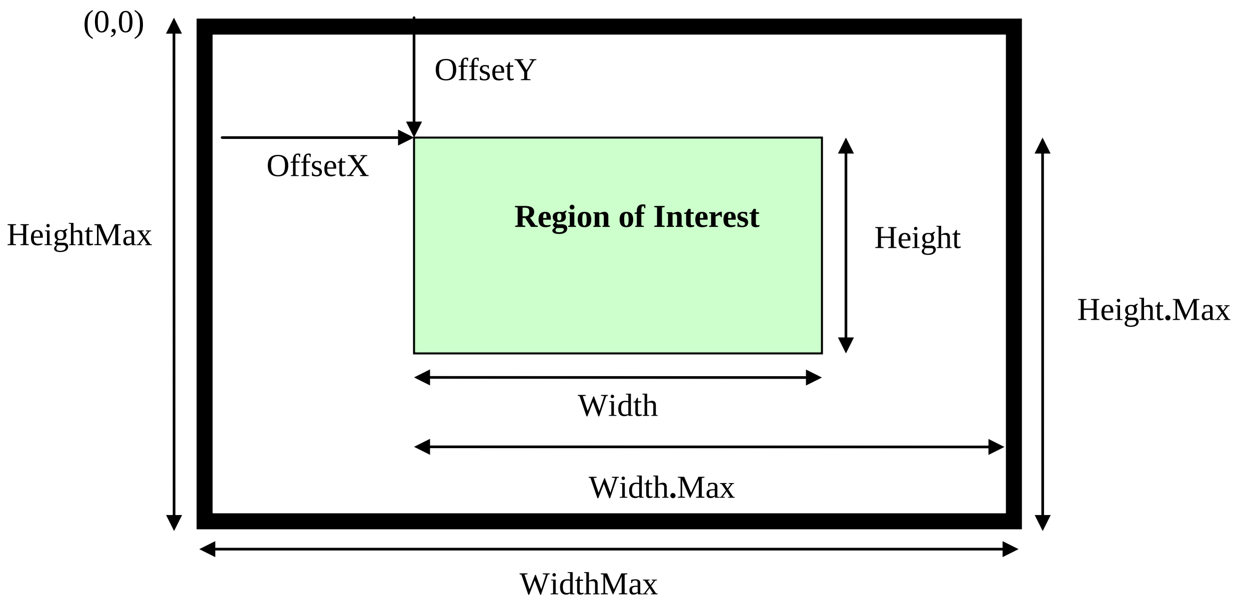


Figure 4-1: Image size and defining a Region of interest

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

Using **BinningHorizontal** and/or **BinningVertical** or **DecimationHorizontal** and/or **DecimationVertical** the image is shrunk to**WidthMax** times **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 Region of interest is applied.

Within the shrunk image the user can set a Region of interest using the features **OffsetX**, **OffsetY**, **Width**, and **Height**. The resulting image generated by the device 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. If multiple Regions of interest are supported by the device, the **RegionSelector**, **RegionMode** and **RegionDestination** features can be used to select and control each Region individually.All measures are given in 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 the **PixelFormat** featur**e**. Only a subset of the possible pixel formats is presented in this document. The complete list of possible standard pixel formats and their layout can be found in the "Pixel Format naming Convention (PFNC) " specification hosted by the AIA organization (see http://www.visiononline.org). .Because the **PixelFormat** feature contains a mix of informations specified by the user and informations provided by the device, 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, **PixelDynamicRangeMin** and **PixelDynamicRangeMax** specify the lower and upper limits of the pixel values in the image. In general, **PixelDynamicRangeMin** should be zero and **PixelDynamicRangeMax** should be a power of two ($\left[0, 2^{DataDepth} - 1\right]$). There should be no missing codes in the range.

4.1 ImageFormatControl

Name	ImageFormatControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category for Image Format Control features.

4.2 SensorWidth

Name	SensorWidth
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	>0

Effective width of the sensor in pixels.

4.3 SensorHeight

Name	SensorHeight
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	>0

Effective height of the sensor in pixels.

For linescan sensor, this value is 1.

4.4 SensorTaps

Name	SensorTaps
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	One Two Three Four Eight Ten … Device-specific

Number of taps of the camera sensor.

Possible values are:

- One: 1 tap.
- Two: 2 taps.
- Three: 3 taps.
- Four: 4 taps.

- Eight: 8 taps.
- Ten: 10 taps.

4.5 SensorDigitizationTaps

Name	SensorDigitizationTaps
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	One Two Three Four Eight Ten … Device-specific

Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

Possible values are:

- **One**: 1 tap.
- **Two**: 2 taps.
- **Three**: 3 taps.
- **Four**: 4 taps.
- **Eight**: 8 taps.
- **Ten**: 10 taps.

4.6 WidthMax

Name	WidthMax
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert

Values	>0

Maximum width of the image (in pixels). The dimension is calculated after horizontal binning, decimation or any other function changing the horizontal dimension of the image.

WidthMax does not take into account the current Region of interest (Width or OffsetX). Its value must be greater than 0 and less than or equal to SensorWidth (unless an oversampling feature is present).

4.7 HeightMax

Name	HeightMax
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	>0

Maximum height of the image (in pixels). This dimension is calculated after vertical binning, decimation or any other function changing the vertical dimension of the image

HeightMax does not take into account the current Region of interest (Height or OffsetY). Its value must be greater than 0 and less than or equal to SensorHeight (unless an oversampling feature is present).

4.8 RegionSelector

Name	RegionSelector
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	Region0 (if 0 based) Region1 Region2 … All

Selects the Region of interest to control. The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently.

Note that if multiple Regions of interest are supported by the device, the **RegionSelector** can be added to various features such as Width, Height,.. to specify the behavior of the selected Region. In order to simplify the standard text and feature descriptions, the optional **RegionSelector** is not propagated to all the features of the SFNC that it can potentially select.Possible values are:

- **Region0**: Selected feature will control the region 0.
- **Region1**: Selected feature will control the region 1.
- **Region2**: Selected feature will control the region 2.
- **All**: Selected features will control all the regions at the same time.

4.9 RegionMode

Name	RegionMode[RegionSelector]
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
sUnit	-
Visibility	Beginner
Values	Off On

Controls if the selected Region of interest is active and streaming.

Possible values are:

- **Off**: Disable the usage of the Region.
- **On**: Enable the usage of the Region.

4.10 RegionDestination

Name	RegionDestination[RegionSelector]
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Stream0

	Stream1 Stream2 …

Control the destination of the selected region.

Possible values are:

- **Stream0**: The destination of the region is the data stream 0.
- **Stream1**: The destination of the region is the data stream 1.
- **Stream2**: The destination of the region is the data stream 2.
- …

4.11 Width

Name	Width[RegionSelector]
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	>0

Width of the image provided by the device (in pixels).

This reflects the current Region of interest. The maximum value of this feature takes into account horizontal binning, decimation, or any other function changing the maximum horizontal dimensions of the image and is typically equal to WidthMax - OffsetX.

This feature is generaly mandatory for transmitters and transceivers of most Transport Layers.

4.12 Height

Name	Height[RegionSelector]
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	>0

Height of the image provided by the device (in pixels).

This reflects the current Region of interest. The maximum value of this feature takes into account vertical binning, decimation, or any other function changing the maximum vertical dimensions of the image and is typically equal to HeightMax - OffsetY.

This feature is generaly mandatory for transmitters and transceivers of most Transport Layers.

4.13 OffsetX

Name	OffsetX[RegionSelector]
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Beginner
Values	≥0

Horizontal offset from the origin to the region of interest (in pixels).

4.14 OffsetY

Name	OffsetY[RegionSelector]
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Beginner
Values	≥0

Vertical offset from the origin to the region of interest (in pixels).

4.15 LinePitch

Name	LinePitch[RegionSelector]
Category	ImageFormatControl
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	B

Visibility	Expert
Values	≥0

Total number of bytes between 2 successive lines. This feature is used to facilitate alignment of image data.

This might be useful if the system has specific limitations, such as having the lines aligned on 32-bit boundaries.

4.16 BinningSelector

Name	BinningSelector
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Sensor Region0 (if 0 based) Region1 Region2 …

Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

Possible values are:

- **Sensor**: Selected features will control the sensor binning.
- **Region0**: Selected feature will control the region 0 binning.
- **Region1**: Selected feature will control the region 1 binning.
- **Region2**: Selected feature will control the region 2 binning.

4.17 BinningHorizontal

Name	BinningHorizontal[BinningSelector]
Category	ImageFormatControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert

Values	>0

Number of horizontal photo-sensitive cells to combine together. This increases the intensity (or signal to noise ratio) of the pixels and reduces the horizontal resolution (width) of the image.

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

4.18 BinningVertical

Name	BinningVertical[BinningSelector]
Category	ImageFormatControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	>0

Number of vertical photo-sensitive cells to combine together. This increases the intensity (or signal to noise ratio) of the pixels and reduces the vertical resolution (height) of the image.

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

4.19 DecimationHorizontal

Name	DecimationHorizontal
Category	ImageFormatControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Horizontal sub-sampling of the image. This reduces the horizontal resolution (width) of the image by the specified horizontal decimation factor.

This might be done by pixel dropping or by first applying a horizontal low-pass filter before pixel dropping.

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

4.20 DecimationVertical

Name	DecimationVertical
Category	ImageFormatControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Vertical sub-sampling of the image. This reduces the vertical resolution (height) of the image by the specified vertical decimation factor.

This might be implemented by pixel dropping or by first applying a vertical low-pass filter before pixel dropping.

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

4.21 ReverseX

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

Flip horizontally the image sent by the device**.** The Region of interest is applied after the flipping.

4.22 ReverseY

Name	ReverseY
Category	ImageFormatControl
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-

Visibility	Expert
Values	True False

Flip vertically the image sent by the device**.** The Region of interest is applied after the flipping.

4.23 PixelFormat

Name	PixelFormat
Category	ImageFormatControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	Mono1p Mono2p Mono4p Mono8 Mono8s Mono10 Mono10c3a64 Mono10c3p32 Mono10g12 Mono10msb Mono10p Mono10pmsb Mono10s Mono12 Mono12g Mono12msb Mono14 Mono16 R8 G8 B8 RGB8 RGB8_Planar RGB8a32 RGBa8 RGB10 RGB10_Planar RGB10g32 RGB10g32msb

	RGB10p32 RGB10p32msb RGB12 RGB12_Planar RGB16 RGB16_Planar RGB565p BGR10 BGR12 BGR16 BGR565p BGR8 BGRa8 YUV411_8 YUV422_8 YUV8 YCbCr411_8 YCbCr422_8 YCbCr601_411_8 YCbCr601_422_8 YCbCr601_8 YCbCr709_411_8 YCbCr709_422_8 YCbCr709_8 YCbCr8 BayerBG8 BayerGB8 BayerGR8 BayerRG8 BayerBG10 BayerBG10g12 BayerGB10 BayerGB10g12 BayerGR10 BayerGR10g12 BayerRG10 BayerRG10g12 BayerBG12 BayerBG12g BayerGB12 BayerGB12g BayerGR12 BayerGR12g BayerRG12 BayerRG12g

	BayerBG16 BayerGB16 BayerGR16 BayerRG16 Raw16 Raw8 Device-specific - GigE Vision Specific: Mono12Packed BayerGR10Packed BayerRG10Packed BayerGB10Packed BayerBG10Packed BayerGR12Packed BayerRG12Packed BayerGB12Packed BayerBG12Packed RGB10V1Packed BGR10V1Packed RGB12V1Packed - Deprecated: Mono8Signed (Deprecated, use Mono8s) RGB8Packed (Deprecated, use RGB8) BGR8Packed (Deprecated, use BGR8) RGBA8Packed (Deprecated, use RGBa8) BGRA8Packed (Deprecated, use BGRa8) RGB10Packed (Deprecated, use RGB10) BGR10Packed (Deprecated, use BGR10) RGB12Packed (Deprecated, use RGB12) BGR12Packed (Deprecated, use BGR12) RGB16Packed (Deprecated, use RGB16) BGR16Packed (Deprecated, use BGR16) RGB10V2Packed (Deprecated, use RGB10p32) BGR10V2Packed (Deprecated, use BGR10p32) RGB565Packed (Deprecated, use RGB565p) BGR565Packed (Deprecated, use BGR565p) YUV411Packed (Deprecated, use YUV411_8_UYYVYY) YUV422Packed (Deprecated, use YUV422_8_UYVY) YUV444Packed (Deprecated, use YUV8_UYV) YUYVPacked (Deprecated, use YUV422_8) RGB8Planar (Deprecated, use RGB8_Planar) RGB10Planar (Deprecated, use RGB10_Planar) RGB12Planar (Deprecated, use RGB12_Planar)

	RGB16Planar (Deprecated, use RGB16_Planar)

Format of the pixels provided by the device. It represents all the information provided by **PixelCoding, PixelSize**, **PixelColorFilter** combined in a single feature.

Note that only a subset of the possible pixel formats is listed here. The complete list of possible standard pixel formats and their detailed layout can be found in the "Pixel Format Naming Convention (PFNC)" specification hosted by the AIA organisation. Refer to the most recent version of that convention for additional information about the construction of a pixel format name.

Possible values are:

- **Mono1p**: Mono 1 bit packed.
- **Mono2p**: Mono 2 bit packed.
- **Mono4p**: Mono 4 bit packed.
- **Mono8**: Mono 8 bit packed.
- **Mono8s**: Mono 1 bit signed.
- **Mono10**: Mono 10 bit.
- **Mono10c3a64**: Mono 10 bit in 64 bit.
- **Mono10c3p32**: Mono 10 bit in 32bit.
- **Mono10g12**: Mono 10 bit grouped in 12 bit.
- **Mono10msb**: Mono 10 bit packed aligned on Msb.
- **Mono10p**: Mono 10 bit packed.
- **Mono10pmsb**: Mono 10 bit packed aligned on Msb.
- **Mono10s**: Mono 10 bit signed.
- **Mono12**: Mono 12 bit packed.
- **Mono12g**: Mono 12 bit grouped.
- **Mono12msb**: Mono 12 bit aligned on Msb.
- **Mono14**: Mono 14 bit.
- **Mono16**: Mono 16 bit.
- **R8**: Red 8 bit.
- **G8**: Green 8 bit.
- **B8**: Blue 8 bit.
- **RGB8**: Red, Green, Blue 8 bit
- **RGB8_Planar**: Red, Green, Blue 8 bit planar.
- **RGB8a32:** Red, Green, Blue 8 bit aligned in 32 bit pixel
- **RGBa8:** Red, Green, Blue 8 bit aligned on 8 bit
- **RGB10:** Red, Green, Blue 10 bit.

- **RGB10_Planar:** Red, Green, Blue 10 bit planar.
- **RGB10g32:** Red, Green, Blue 8 bit grouped in 32 bit pixel.
- **RGB10g32msb:** Red, Green, Blue 10 bit grouped in 32 bit pixel aligned on Msb.
- **RGB10p32:** Red, Green, Blue 10 bit packed in 32 bit pixel.
- **RGB10p32msb:** Red, Green, Blue 10 bit packed in 32 bit pixel.
- **RGB12:** Red, Green, Blue 12 bit.
- **RGB12_Planar:** Red, Green, Blue 12 bit planar.
- **RGB16:** Red, Green, Blue 16 bit.
- **RGB16_Planar:** Red, Green, Blue 16 bit planar.
- **RGB565p:** Red, Green, Blue 16 bit packet in 5, 6, 5 bits.
- **BGR10:** Blue, Green, Red, 10 bit.
- **BGR12:** Blue, Green, Red, 12 bit.
- **BGR16:** Blue, Green, Red, 16 bit.
- **BGR565p:** Blue, Green, Red, 16 bit packet in 5, 6, 5 bits.
- **BGR8:** Blue, Green, Red, 8 bit.
- **BGRa8:** Blue, Green, Red, Alpha 8 bit.
- **YUV411_8:** YUV 411 8 bit.
- **YUV422_8:** YUV 422 8 bit.
- **YUV8:** YUV 8 bit.
- **YCbCr411_8:** YCrCb 411 8 bit.
- **YCbCr422_8:** YCrCb 422 8 bit.
- **YCbCr601_411_8:** YCrCb 601 411 8 bit.
- **YCbCr601_422_8:** YCrCb 601 422 8 bit.
- **YCbCr601_8:** YCrCb 601 8 bit.
- **YCbCr709_411_8:** YCrCb 709 411 8 bit.
- **YCbCr709_422_8:** YCrCb 709 422 8 bit.
- **YCbCr709_8:** YCrCb 709 8 bit.
- **YCbCr8:** YCbCr 8 bit.
- **BayerBG8:** Bayer Blue Green 8 bit.
- **BayerGB8:** Bayer Green Blue 8 bit.
- **BayerGR8:** Bayer Green Red 8 bit.
- **BayerRG8:** Bayer Red Green 8 bit.
- **BayerBG10:** Bayer Blue Green 10 bit.

- **BayerBG10g12:** Bayer Blue Green 8 bit grouped on 12 bit.
- **BayerGB10:** Bayer Green Blue 10 bit.
- **BayerGB10g12:** Bayer Green Blue 10 bit grouped on 12 bit.
- **BayerGR10:** Bayer Green Red 10 bit.
- **BayerGR10g12:** Bayer Green Red 10 bit grouped on 12 bit.
- **BayerRG10:** Bayer Red Green 10 bit.
- **BayerRG10g12:** Bayer Red Green 10 bit grouped on 12 bit.
- **BayerBG12:** Bayer Blue Green 12 bit
- **BayerBG12g:** Bayer Blue Green 12 bit grouped.
- **BayerGB12:** Bayer Green Blue 12 bit
- **BayerGB12g:** Bayer Green Blue 12 bit grouped on 12 bit.
- **BayerGR12:** Bayer Green Red 12 bit.
- **BayerGR12g:** Bayer Green Red 12 bit grouped on 12 bit.
- **BayerRG12:** Bayer Red Green 12 bit.
- **BayerRG12g:** Bayer Red Green 12 bit grouped on 12 bit.
- **BayerBG16:** Bayer Blue Green 16 bit.
- **BayerGB16:** Bayer Green Blue 16 bit.
- **BayerGR16:** Bayer Green Red 16 bit.
- **BayerRG16:** Bayer Red Green 16 bit.
- **Raw16:** Raw 16 bit.
- **Raw8:** Raw bit.
- **Mono12Packed:** Mono 12 bit packed (GigE Vision Specific).
- **BayerGR10Packed:** Bayer GR 10 bit packed (GigE Vision Specific).
- **BayerRG10Packed:** Bayer RG 10 bit packed (GigE Vision Specific).
- **BayerGB10Packed:** Bayer GB 10 bit packed (GigE Vision Specific).
- **BayerBG10Packed:** Bayer BG 10 bit packed (GigE Vision Specific).
- **BayerGR12Packed:** Bayer GR 12 bit packed (GigE Vision Specific).
- **BayerRG12Packed:** Bayer RG 12 bit packed (GigE Vision Specific).
- **BayerGB12Packed:** Bayer GB 12 bit packed (GigE Vision Specific).
- **BayerBG12Packed:** Bayer BG 12 bit packed (GigE Vision Specific).
- **RGB10V1Packed:** RGB 10 bit packed (GigE Vision Specific).
- **BGR10V1Packed:** BGR 10 bit packed (GigE Vision Specific).
- **RGB12V1Packed:** RGB 12 bit packed (GigE Vision Specific).

The values of the enumeration entries are transport layer specific. Refer to the appropriate transport layer specification for additional information.

This feature is generaly mandatory for transmitters and transceivers of most Transport Layers.

4.24 PixelFormatInfoSelector

Name	PixelFormatInfoSelector
Category	ImageFormatControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	Mono1p Mono2p Mono4p Mono8 Mono8s Mono10 …

Select the pixel format for which the information will be returned.

The pixel format selected must be one of the values present in the PixelFormat feature.

Possible values are:

- **Mono1p**: Mono 1 bit packed.
- **Mono2p**: Mono 2 bit packed.
- **Mono4p**: Mono 4 bit packed.
- **Mono8**: Mono 8 bit packed.
- **Mono8s**: Mono 1 bit signed.
- **Mono10**: Mono 10 bit.
- **…**

Note: This feature must be a floating node and should always be available.

4.25 PixelFormatInfoID

Name	PixelFormatInfoID[PixelFormatInfoSelector]
Category	ImageFormatControl
Level	Recommended
Interface	IInteger

Access	Read
Unit	-
Visibility	Guru
Values	-

Returns the value used by the streaming channels to identify the selected pixel format.

To change the Pixel format that will be sent by the device, the PixelFormat feature should be used.

Note: This feature must be a floating node and should always be available.

4.26 PixelCoding (Deprecated)

Name	PixelCoding
Category	ImageFormatControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Mono MonoSigned MonoPacked RGBPacked BGRPacked RGBAPacked BGRAPacked RGBPlanar YUV411Packed YUV422Packed YUV444Packed YUYVPacked Raw RawPacked

This feature is deprecated. It represents the coding of the pixels in the image. Raw gives the data in the native format of the sensor.

Possible values are:

- **Mono**: Mono.
- **MonoSigned:** Mono signed.
- **MonoPacked**: Mono packed.
- **RGBPacked**: RGB packed.
- **BGRPacked**: BGR packed.

- **RGBAPacked**: RGBA packed.
- **BGRAPacked**: BGRA packed.
- **RGBPlanar**: RGB planar.
- **YUV411Packed**: YUV 411 packed.
- **YUV422Packed**: YUV 422 packed.
- **YUV444Packed**: YUV 444 packed.
- **YUYVPacked**: YUYV packed.
- **Raw**: Raw.
- **RawPacked**: Raw packed.

Raw is mainly used for Bayer sensor. This value must always be coherent with the **PixelFormat** feature.

4.27 PixelSize

Name	PixelSize
Category	ImageFormatControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Bpp1 Bpp2 Bpp4 Bpp8 Bpp10 Bpp12 Bpp14 Bpp16 Bpp24 Bpp30 Bpp32 Bpp36 Bpp48 Bpp64

Total size in bits of a pixel of the image.

This value must always be coherent with the **PixelFormat** feature.

Possible values are:

- **Bpp1**: 1 bit per pixel.

- **Bpp2**: 2 bits per pixel.
- **Bpp4**: 4 bits per pixel.
- **Bpp8**: 8 bits per pixel.
- **Bpp10**: 10 bits per pixel.
- **Bpp12:** 12 bits per pixel.
- **Bpp14**: 14 bits per pixel.
- **Bpp16**: 16 bits per pixel.
- **Bpp24**: 24 bits per pixel.
- **Bpp30**: 30 bits per pixel.
- **Bpp32**: 32 bits per pixel.
- **Bpp36**: 36 bits per pixel.
- **Bpp48**: 48 bits per pixel.
- **Bpp64**: 64 bits per pixel.

4.28 PixelColorFilter

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

Type of color filter that is applied to the image.

This value must always be coherent with the **PixelFormat** feature.

Possible values are:

- **None**: No color filter.
- **BayerRG**: Bayer Red Green filter.
- **BayerGB**: Bayer Green Blue filter.
- **BayerGR**: Bayer Green Red filter.
- **BayerBG**: Bayer Blue Green filter.

4.29 PixelDynamicRangeMin

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

Minimum 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.

4.30 PixelDynamicRangeMax

Name	PixelDynamicRangeMax
Category	ImageFormatControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	Device-specific

Maximum 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.

4.31 TestPatternGeneratorSelector

Name	TestPatternGeneratorSelector
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Beginner

Values	Sensor Region0 (if 0 based) Region1 Region2 …

Selects which test pattern generator is controlled by the TestPattern feature.

Possible values are:

- **Sensor**: TestPattern feature will control the sensor`s test pattern generator.
- **Region0**: TestPattern feature will control the region 0 test pattern generator.
- **Region1**: TestPattern feature will control the region 1 test pattern generator.
- **Region2**: TestPattern feature will control the region 2 test pattern generator.

4.32 TestPattern

Name	TestPattern[TestPatternGeneratorSelector]
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Off Black White GreyHorizontalRamp GreyVerticalRamp GreyHorizontalRampMoving GreyVerticalRampMoving HorizontalLineMoving VerticalLineMoving ColorBar FrameCounter Device-specific

Selects the type of test pattern that is generated by the device as image source.

Possible values are:

- **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.
- **GreyVerticalRampMoving**: 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.

4.33 TestImageSelector (Deprecated)

Name	TestImageSelector
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Invisible
Values	Off Black White GreyHorizontalRamp GreyVerticalRamp GreyHorizontalRampMoving GreyVerticalRampMoving HorizontalLineMoving VerticalLineMoving ColorBar FrameCounter Device-specific

This feature is deprecated (see TestPattern). Selects the type of test image that is sent by the device.

To help backward compatibility, this feature can be included as Invisible in the device`s XML.

Possible values are:

- **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.
- **GreyVerticalRampMoving**: 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.

4.34 Deinterlacing

Name	Deinterlacing
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Off LineDuplication Weave … Device-specific

Controls how the device performs de-interlacing.

Possible values are:

- **Off**: The device doesn`t perform de-interlacing.
- **LineDuplication**: The device performs de-interlacing by outputting each line of each field twice.
- **Weave**: The device performs de-interlacing by interleaving the lines of all fields.

4.35 Image Compression

This section describes the feature related to image compression.

4.35.1 ImageCompressionMode

Name	ImageCompressionMode
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Visibility	Beginner
Unit	-
Values	Off JPEG JPEG2000 H264 … Device-specific

Enable a specific image compression mode as the base mode for image transfer. Optionally, chunk data can be appended to the compressed image (see chunk section).

Possible values are:

- **Off**: Default value. Image compression is disabled. Images are transmitted uncompressed.
- **JPEG**: JPEG compression is selected.
- **JPEG2000**: JPEG 2000 compression is selected.
- **H264**: H.264 compression is selected.

4.35.2 ImageCompressionRateOption

Name	ImageCompressionRateOption
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration

Access	Read/Write
Unit	-
Visibility	Expert
Values	FixBitrate FixQuality … Device-specific

Two rate controlling options are offered: fixed bit rate or fixed quality. The exact implementation to achieve one or the other is vendor-specific.

Note that not all compression techniques or implementations may support this feature.

Possible values are:

- **FixBitrate**: Output stream follows a constant bit rate. Allows easy bandwidth management on the link.
- **FixQuality**: Output stream has a constant image quality. Can be used when image processing algorithms are sensitive to image degradation caused by excessive data compression.

4.35.3 ImageCompressionQuality

Name	ImageCompressionQuality
Category	ImageFormatControl
Level	Optional
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Device-specific

Control the quality of the produced compressed stream.

This feature is available when ImageCompressionRateOption is equal to FixQuality or if the device only supports the FixQuality mode.

The list of valid values is device-specific. A higher value means a better quality for the produced compressed stream.

4.35.4 ImageCompressionBitrate

Name	ImageCompressionBitrate
Category	ImageFormatControl
Level	Optional
Interface	IFloat

Access	Read/(Write)
Unit	Mbps
Visibility	Expert
Values	Device-specific

Control the rate of the produced compressed stream.

This feature is available when ImageCompressionRateOption is equals to FixBitrate or if the device only supports the FixBitrate mode.

The list of valid values is device specific.

4.35.5 ImageCompressionJPEGFormatOption

Name	ImageCompressionJPEGFormatOption
Category	ImageFormatControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Lossless BaselineStandard BaselineOptimized Progressive … Device-specific

When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

Possible values are:

- **Lossless**: Selects lossless JPEG compression based on a predictive coding model.
- **BaselineStandard**: Indicates this is a baseline sequential (single-scan) DCT-based JPEG.
- **BaselineOptimized**: Provides optimized color and slightly better compression than baseline standard by using custom Huffman tables optimized after statistical analysis of the image content.
- **Progressive**: Indicates this is a progressive (multi-scan) DCT-based JPEG.

5 Acquisition Control

The Acquisition Control chapter 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.

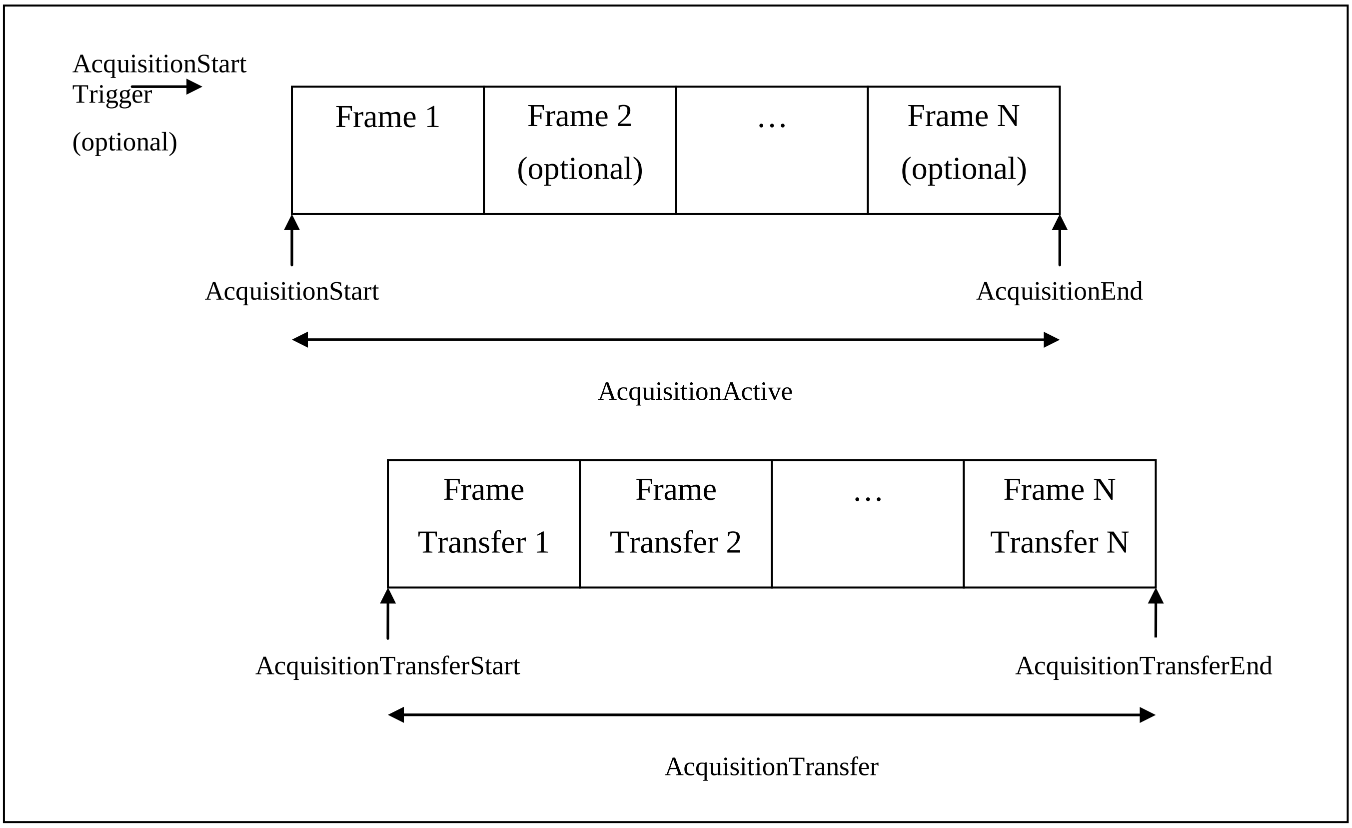
5.1 Acquisition related vocabulary and signals

This section describes the vocabulary and terms used to describe and name the acquisition related features. It also defines the acquisition related signals and their position in time during the acquisition of images by a device.

An **Acquisition** is composed of one or many **Frames** made of **Line(s)**. The **Frames** of an **Acquisition** can optionally be grouped in smaller **Bursts** that are triggered individually. 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 the completion of the transfer of the last one.

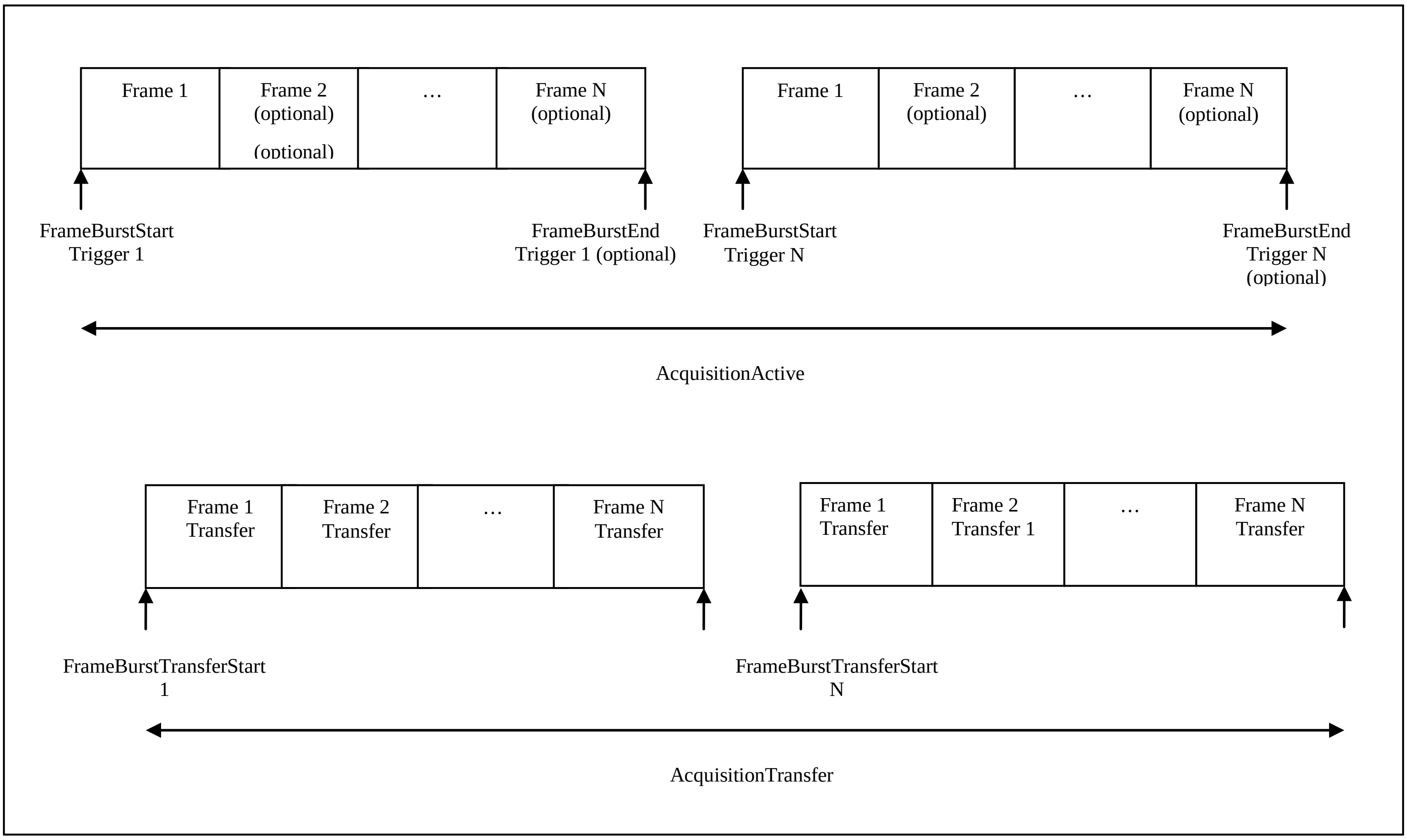
Figure 5-1: Acquisition signals definition



A **Burst** of **Frame**(s) is defined as the capture of a group of one or many **Frame**(s) within an **Acquisition** (see Figure 5-2)**.** If a **FrameBurstStart** or **FrameBurstActive** trigger is enabled (its TriggerMode=On), an acquisition can be broken in many smaller **Bursts.** In this case, each **Burst** has its own trigger. If only the **FrameBurstStart** trigger is enabled, **AcquisitionBurstFrameCount** determines the length of each burst. If the **FrameBurstStart** and **FrameBurstEnd** triggers are enabled, they are used to delimit the length of every single burst. If the **FrameBurstActive** trigger is enabled, it determines the length of each individual burst (the burst lasts as long as the trigger is asserted).

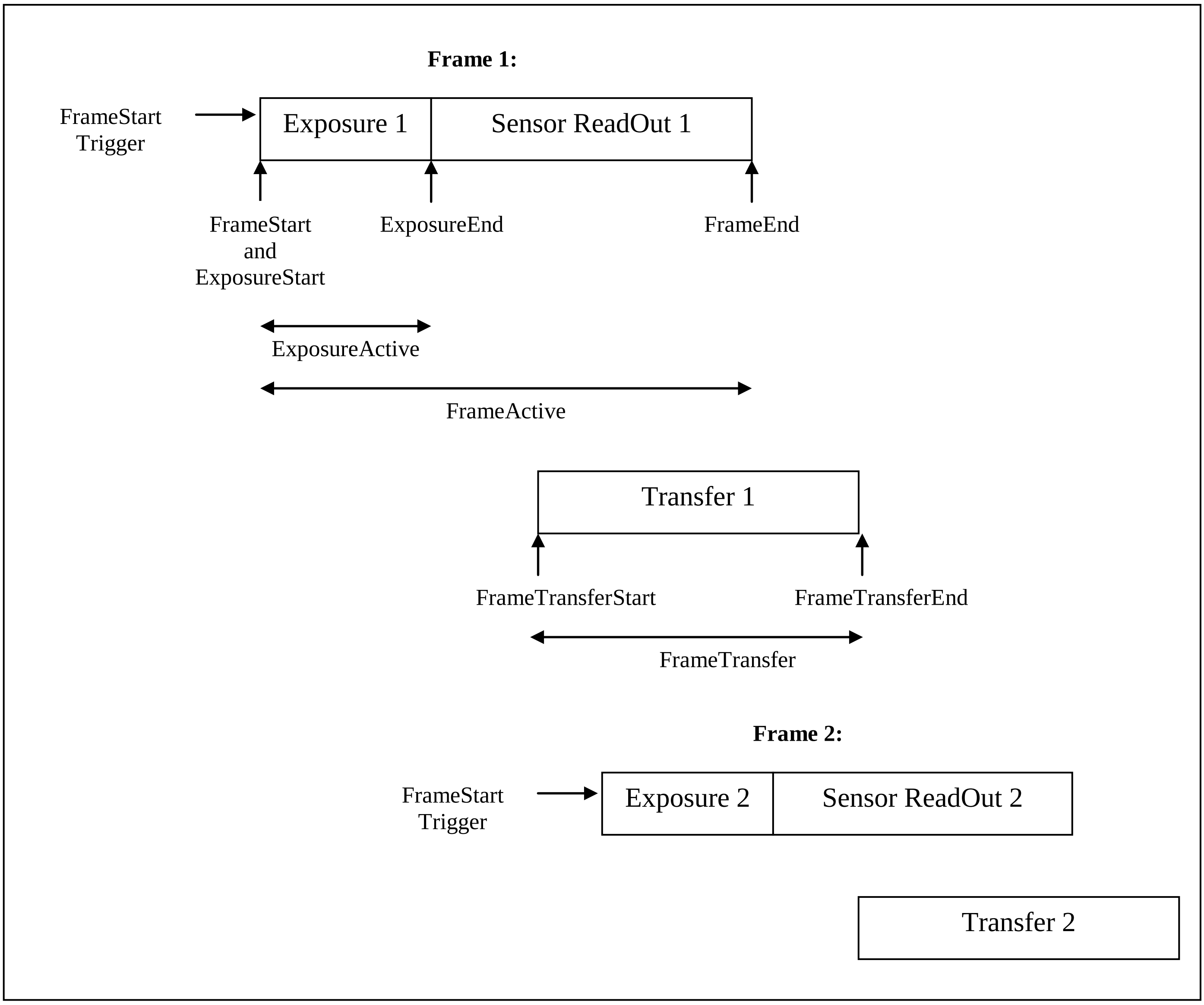
The transfer of the frame(s) of a burst starts with the beginning of the transfer of the first frame of the burst and end with the completion of the transfer of the last one.

Figure 5-2: Burst signals definition



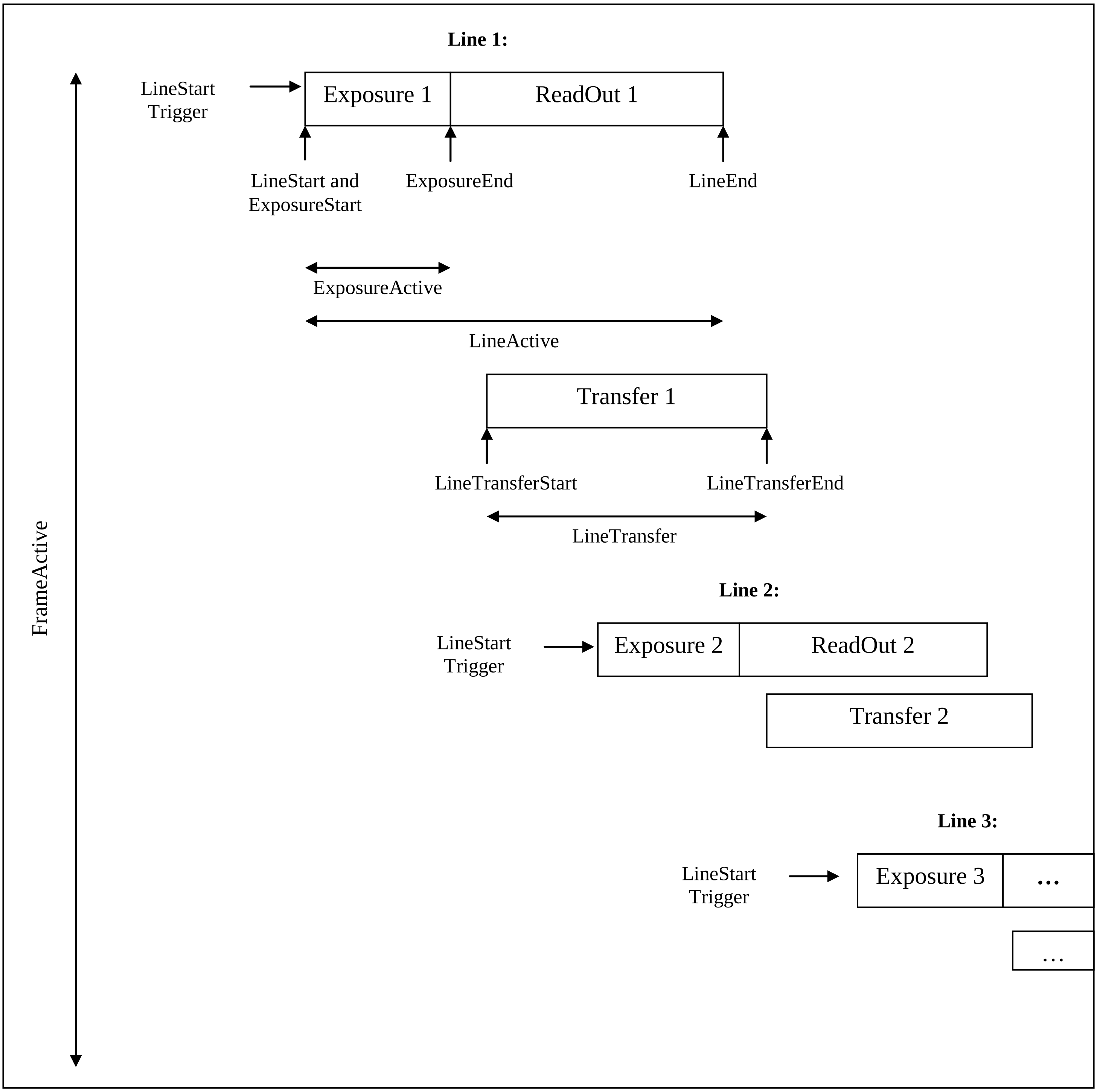
A **Frame** is defined as the capture of **Width** pixels x **Height** lines. A **Frame** starts with an optional **Exposure** period and ends with the completion of the sensor read out. Generally, a transfer period will start during the sensor read out and will finish sometime after it but it is not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

Figure 5-3: Frame signals definition



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.

Figure 5-4: Frame signals definitions in Line Scan mode



5.2 Acquisition features usage model

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

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**.

AcquisitionBurstFrameCount determines the length of each burst to capture if the **FrameBurstStart** trigger is enabled and the **FrameBurstEnd** trigger is disabled.

AcquisitionFrameRate controls the rate at which the Frames are captured when **TriggerMode** is **Off**.

AcquisitionLineRate controls the rate at which the Lines in each Frame are captured when **TriggerMode** is **Off**. 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, ExposureActive** (see Figure 5-1 and Figure 5-3).

See the **Acquisition and Trigger Examples** section of this document for more complete use cases of the acquisition and trigger features in conjunction with other related sections such as I/O and analog controls.

5.3 Acquisition timing diagrams

This section gives the timing diagrams and features setting order for the most common acquisition scenarios.

Figure 5-5: Continuous Acquisition

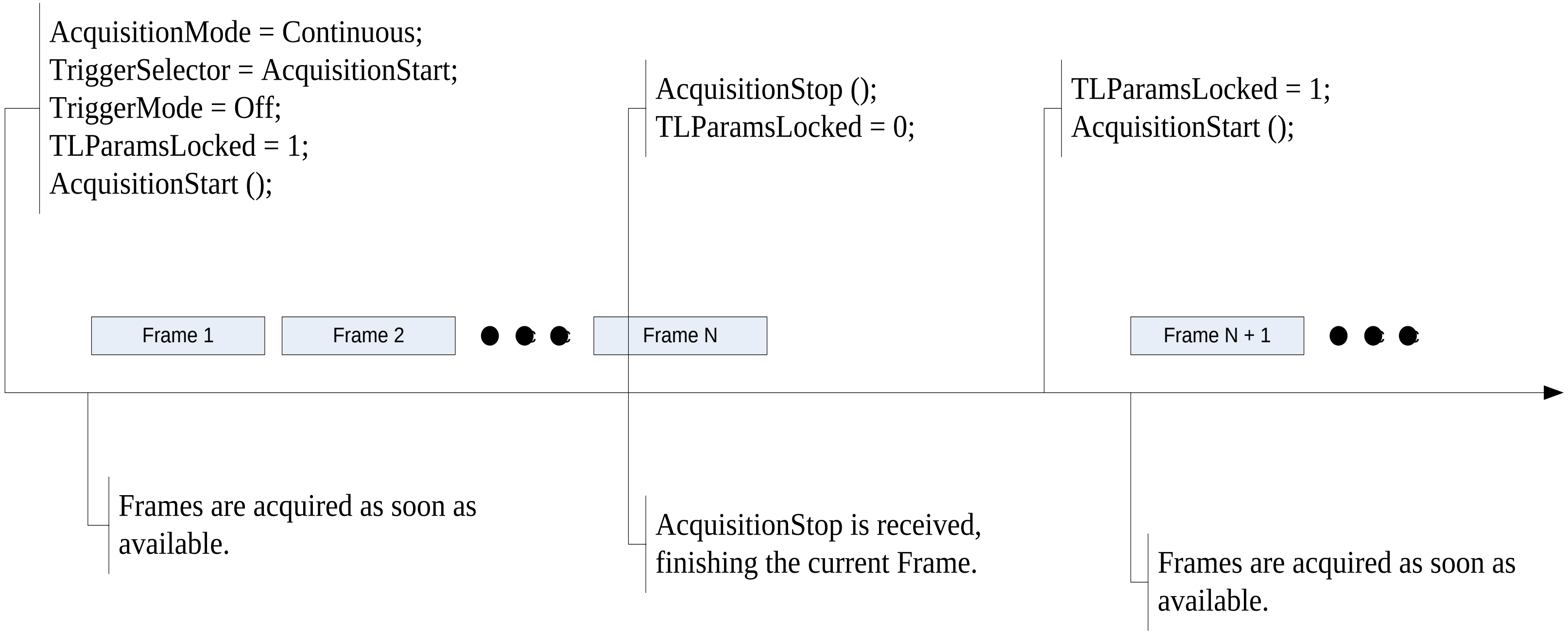


Figure 5-6: Continuous Acquisition with AcquisitionStart trigger

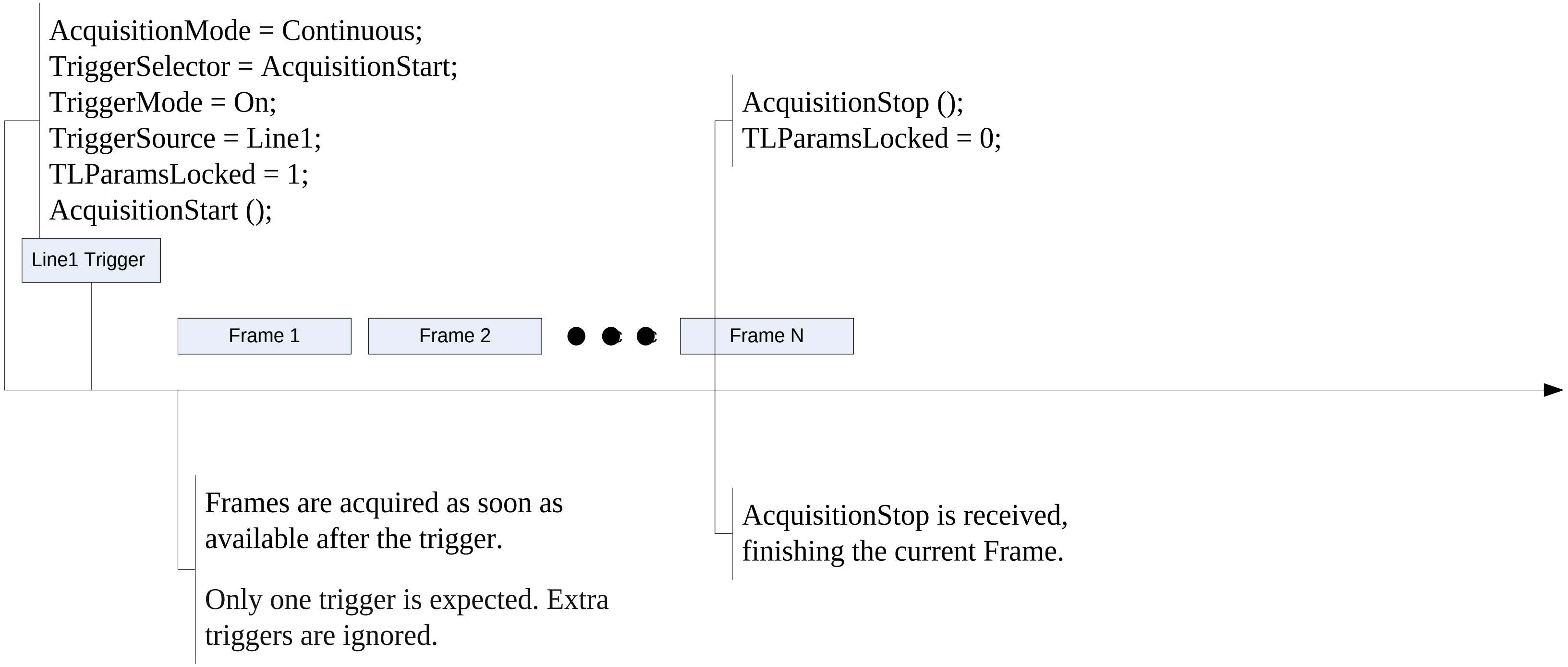


Figure 5-7: Continuous Acquisition with FrameStart trigger

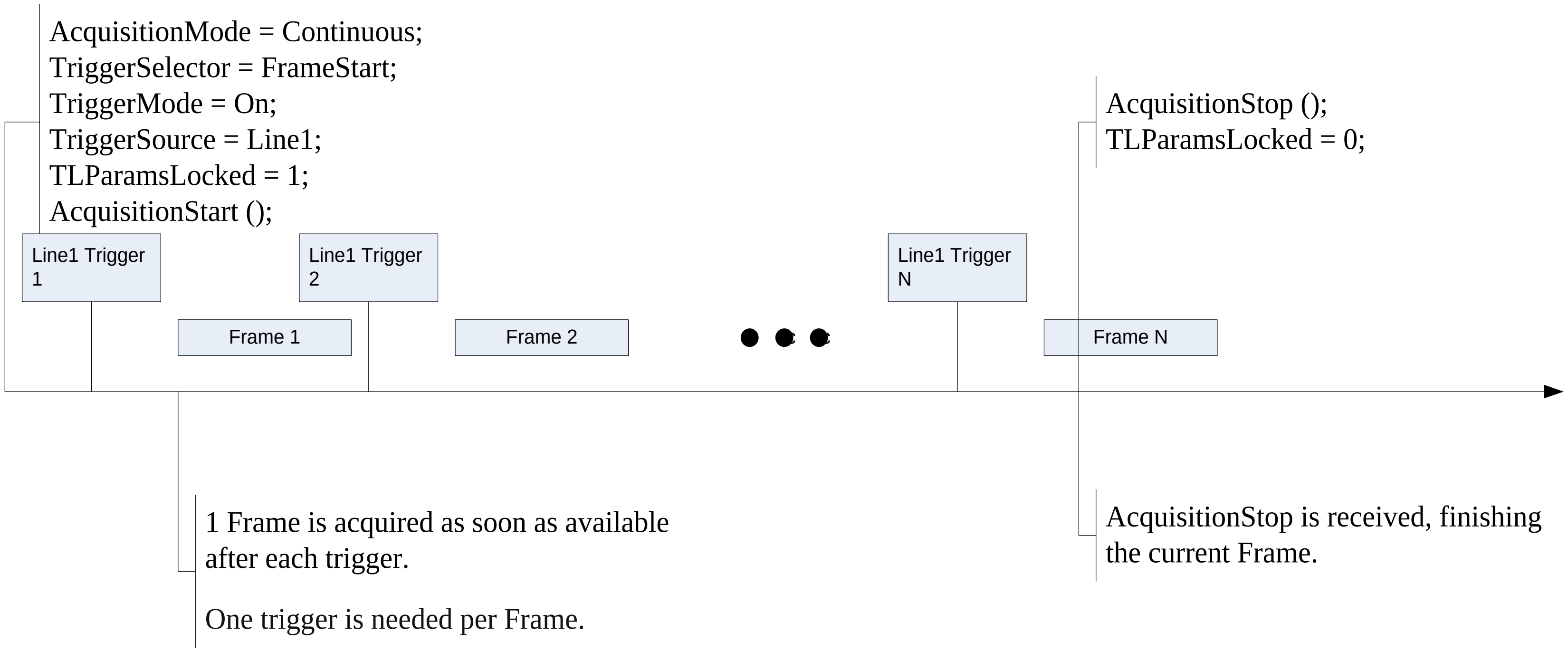


Figure 5-8: Continuous Acquisition with FrameBurstStart trigger

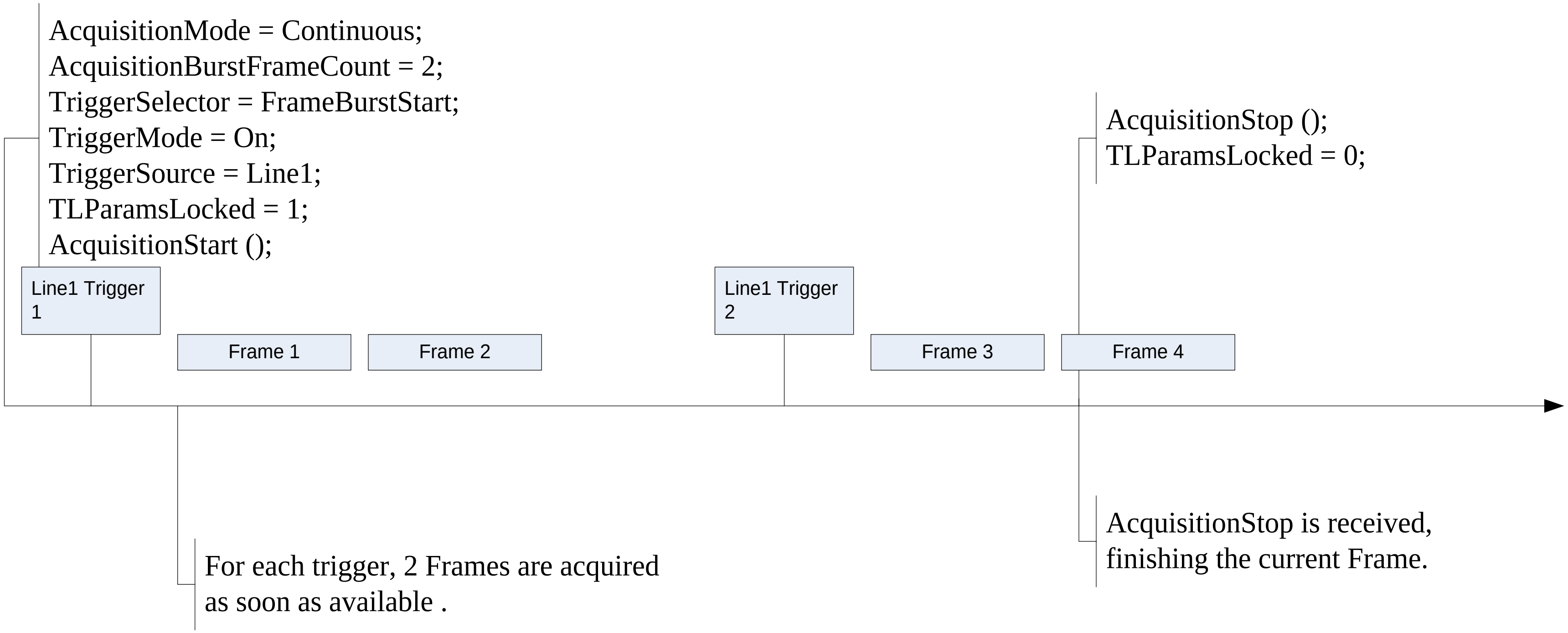


Figure 5-9: MultiFrame Acquisition

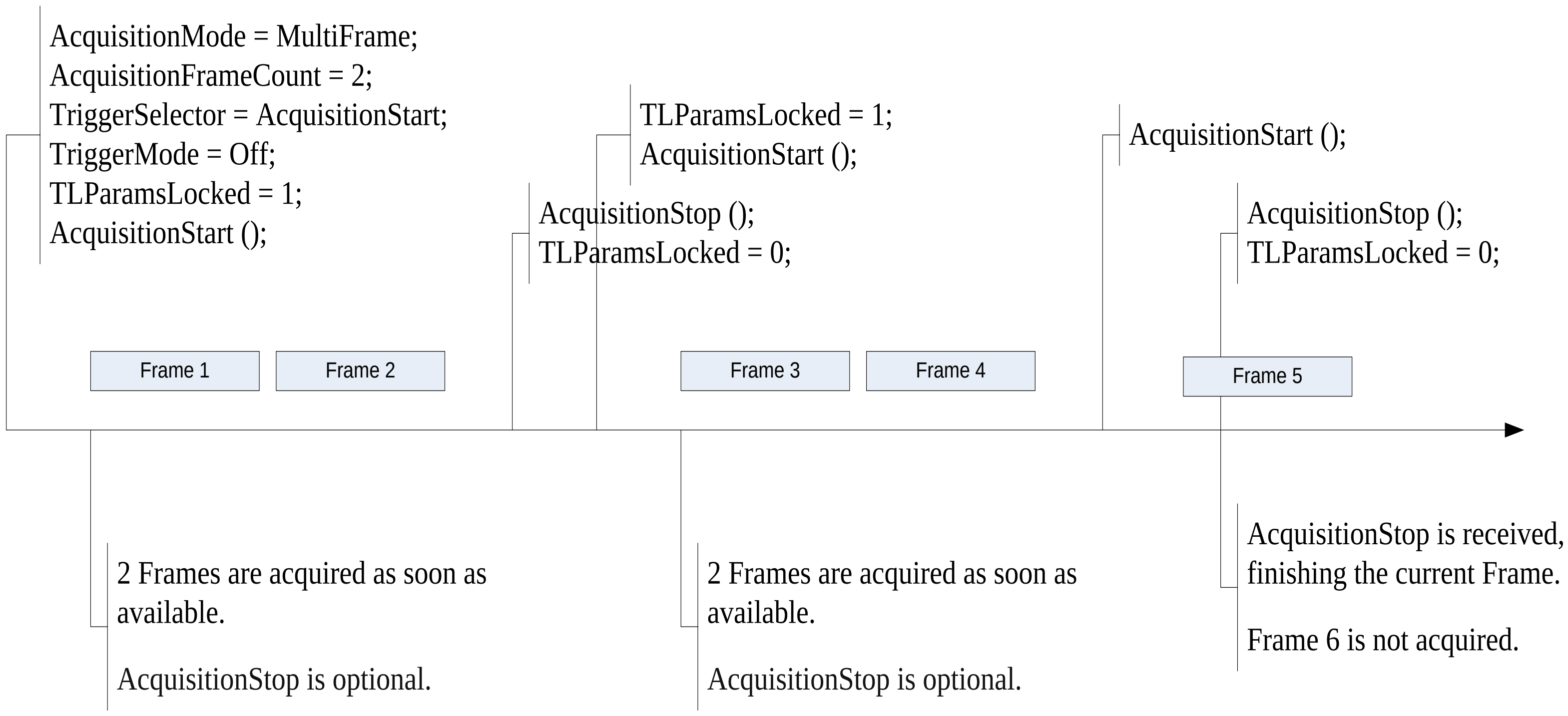


Figure 5-10: Multi Frame Acquisition with AcquisitionStart trigger

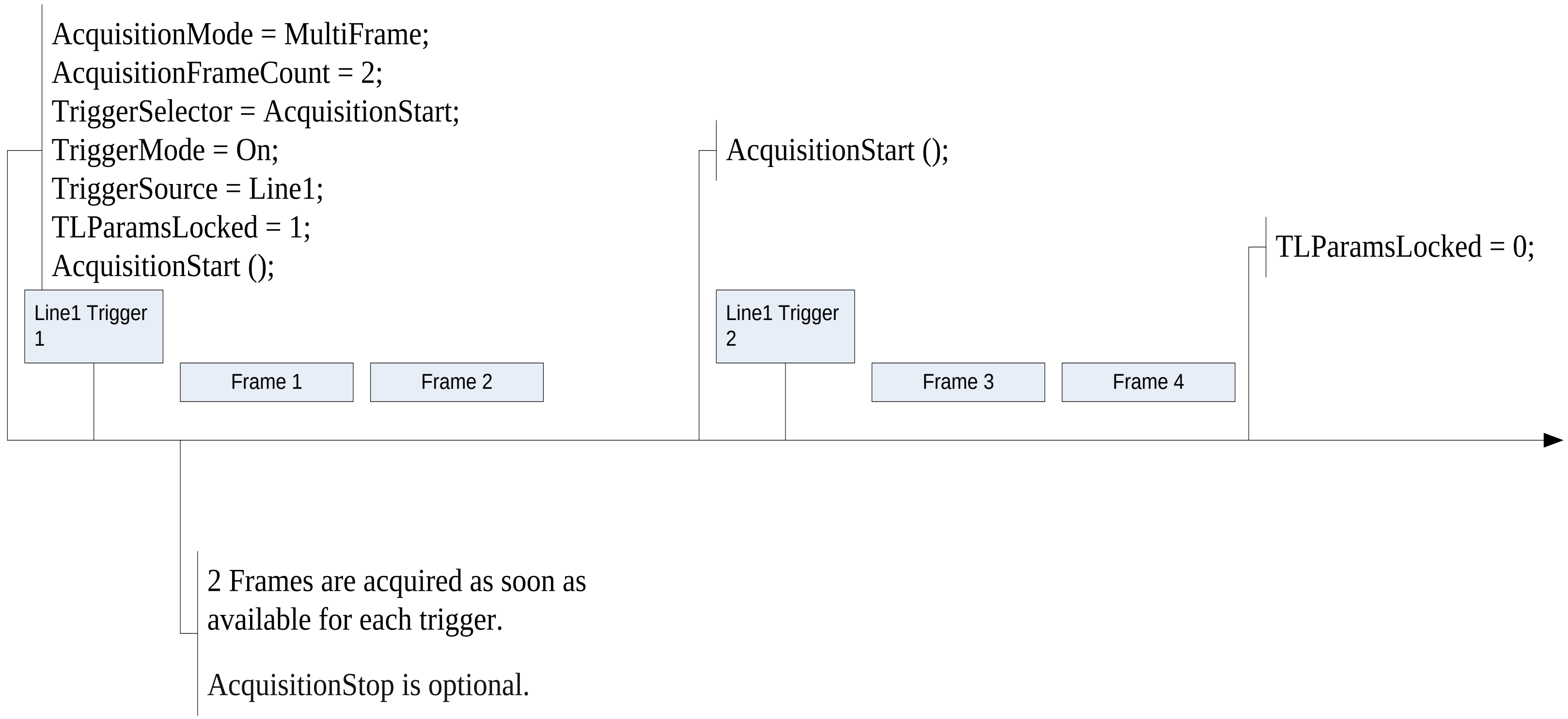


Figure 5-11: Multi Frame Acquisition with FrameStart trigger

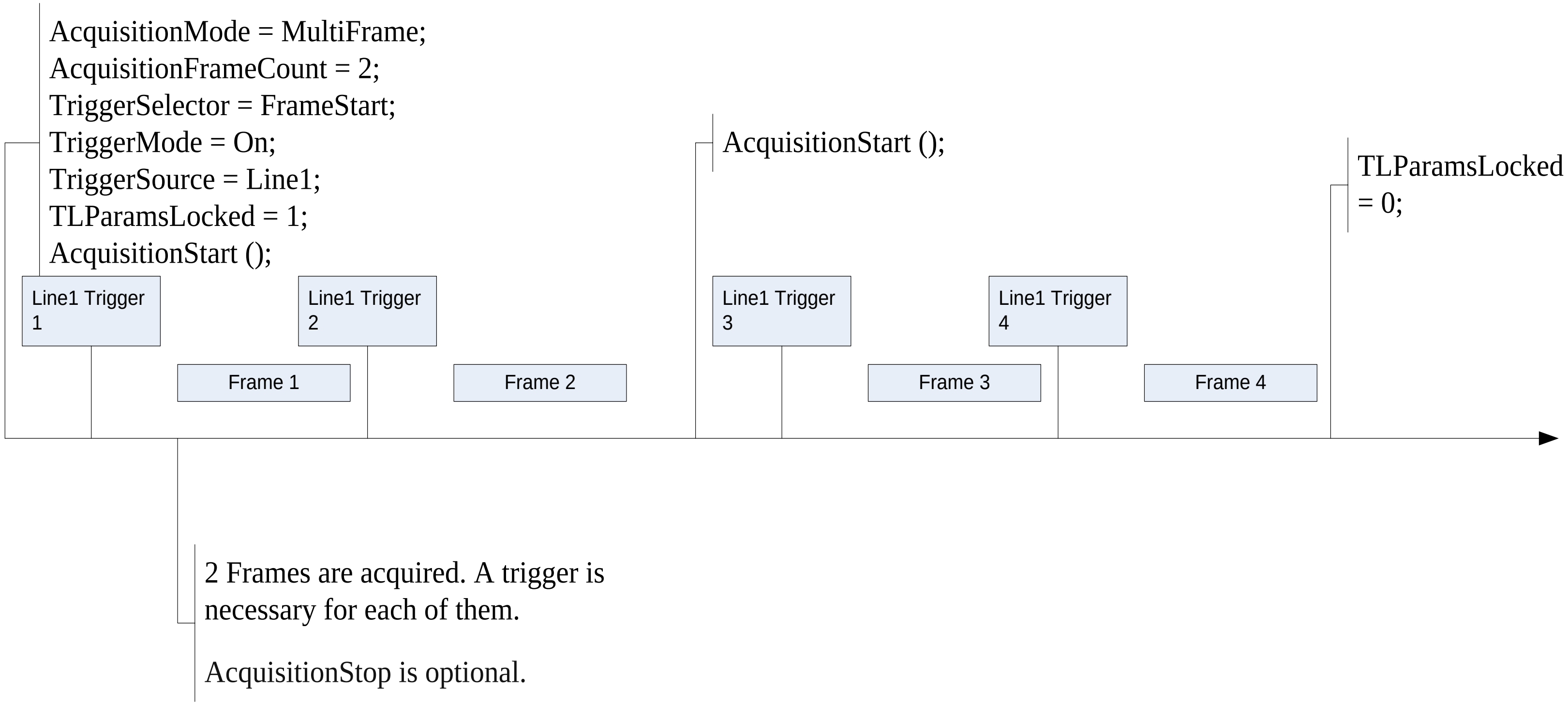


Figure 5-12: Multi Frame Acquisition with FrameBurstStart trigger

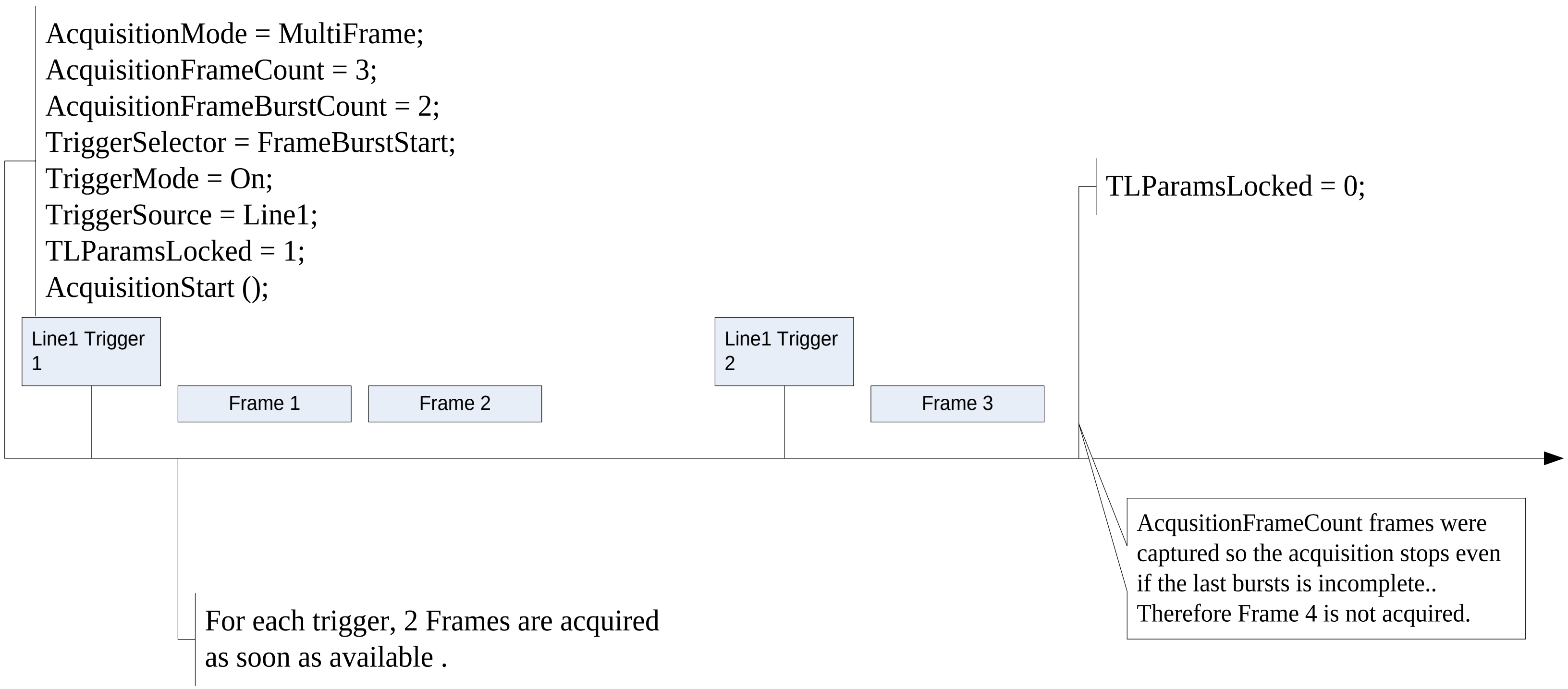


Figure 5-13: Single Frame Acquisition

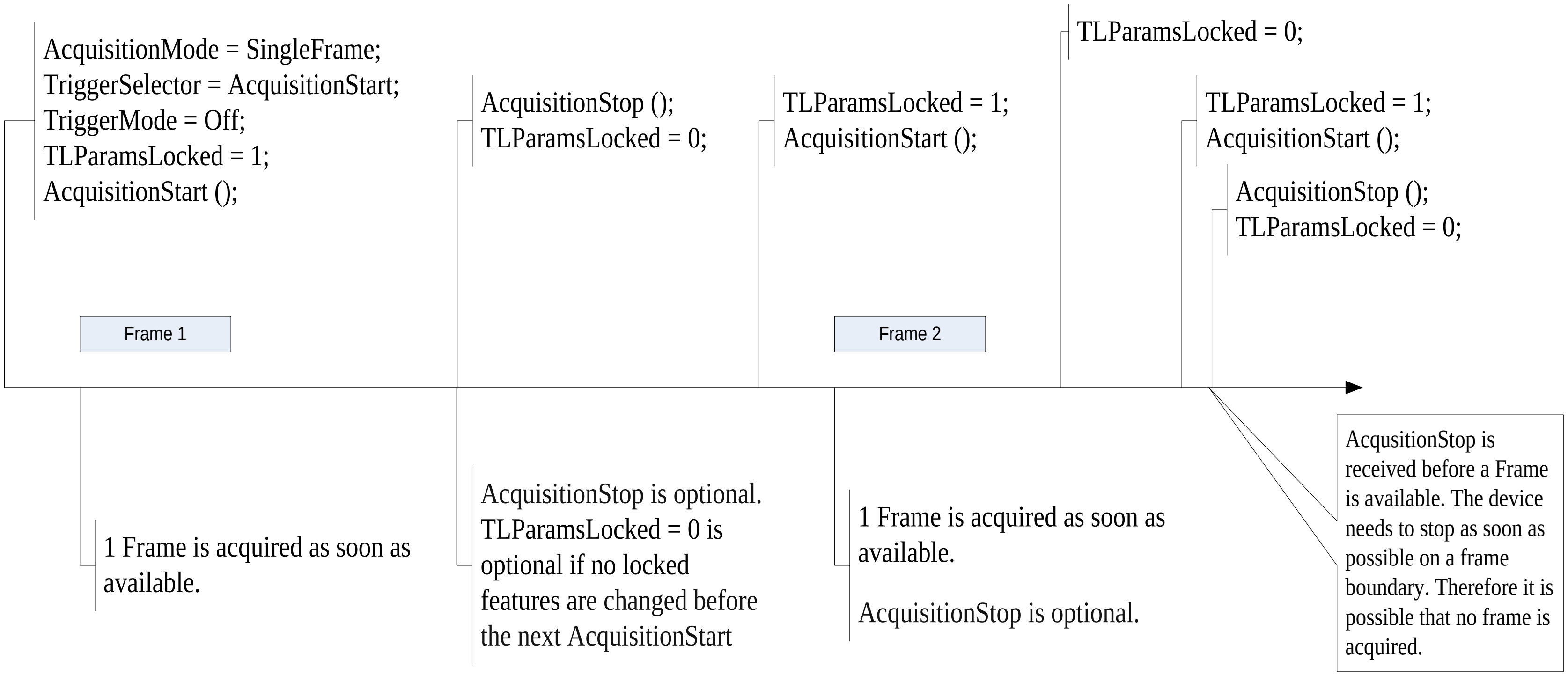


Figure 5-14: Single Frame Acquisition with AcquisitionStart trigger

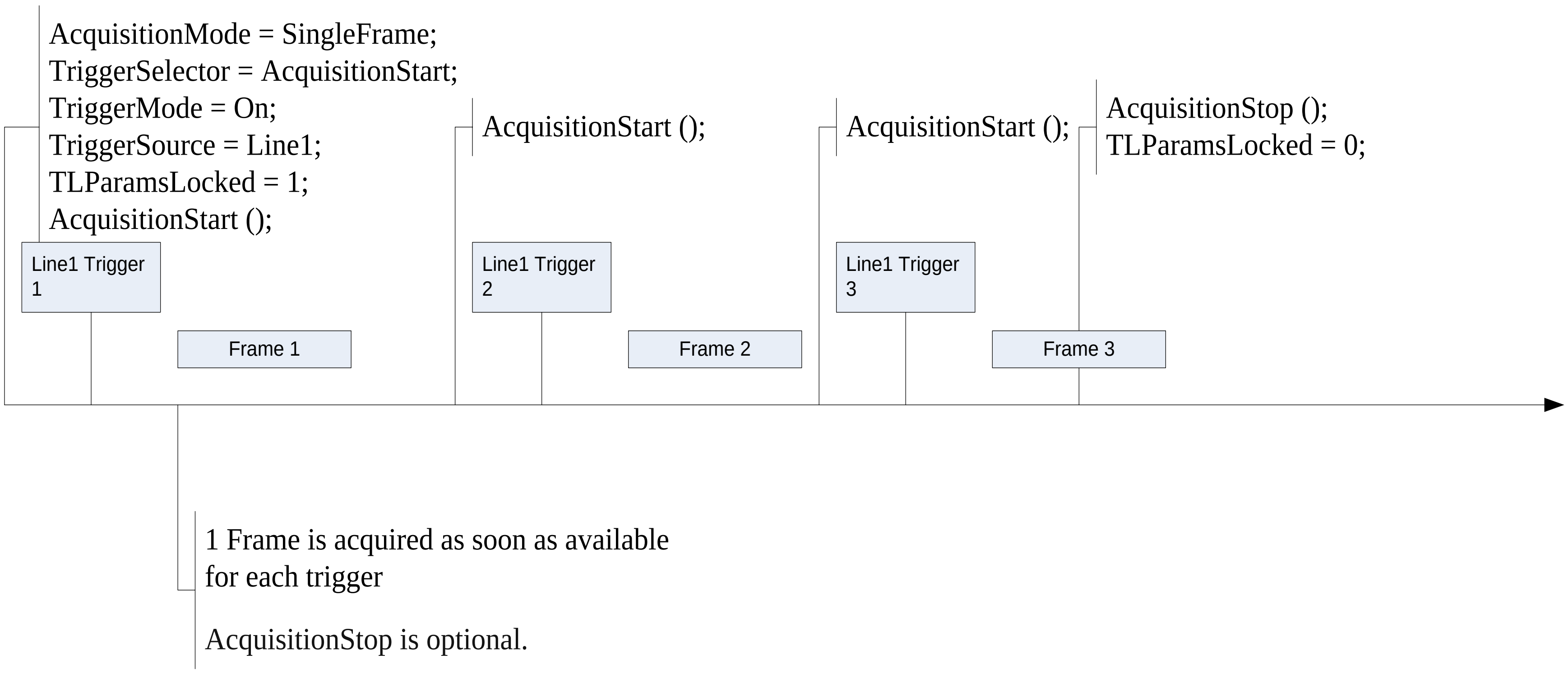
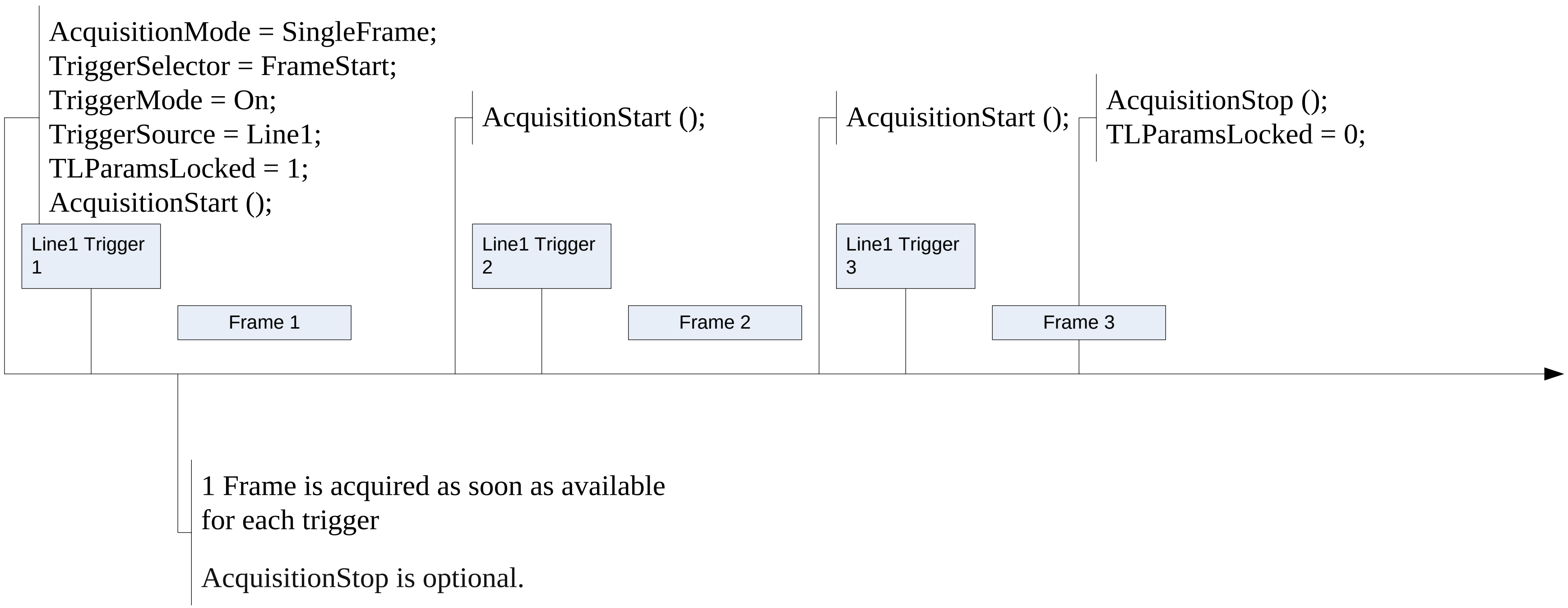


Figure 5-15: Single Frame Acquisition with FrameStart trigger



5.4 Acquisition Control features

This section gives the detailed description of all the Acquisition related features.

5.4.1 AcquisitionControl

Name	AcquisitionControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category for the acquisition and trigger control features.

5.4.2 AcquisitionMode

Name	AcquisitionMode
Category	AcquisitionControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	SingleFrame MultiFrame Continuous

Sets the acquisition mode of the device. It defines mainly the number of frames to capture during an acquisition and the way the acquisition stops.

Possible values are:

- **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.

This feature is generaly mandatory for transmitters and transceivers of most Transport Layers.

5.4.3 AcquisitionStart

Name	AcquisitionStart
Category	AcquisitionControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Beginner
Values	-

Starts the Acquisition of the device. The number of frames captured is specified by **AcquisitionMode.**

The Acquisition might be conditioned by various triggers (see **Trigger**… features). An AcquisitionStart command must be sent to the device before the acquisition related triggers become effective.

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.

If the AcquisitionStart feature is currently not writable (locked), the application must not start the acquisition and must avoid using the feature until the feature becomes writable again.

This feature is generaly mandatory for transmitters and transceivers of most Transport Layers.

5.4.4 AcquisitionStop

Name	AcquisitionStop
Category	AcquisitionControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write

Unit	-
Visibility	Beginner
Values	-

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.

If the camera is waiting for a trigger, the pending Frame will be cancelled. If no Acquisition is in progress, the command is ignored.

This feature is generaly mandatory for transmitters and transceivers of most Transport Layers.

5.4.5 AcquisitionAbort

Name	AcquisitionAbort
Category	AcquisitionControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Aborts the Acquisition immediately. This will end the capture without completing the current Frame or waiting on a trigger. If no Acquisition is in progress, the command is ignored.

5.4.6 AcquisitionArm

Name	AcquisitionArm
Category	AcquisitionControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Arms 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.

5.4.7 AcquisitionFrameCount

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

Number of frames to acquire in MultiFrame Acquisition mode.

5.4.8 AcquisitionBurstFrameCount

Name	AcquisitionBurstFrameCount
Category	AcquisitionControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Beginner
Values	≥1

Number of frames to acquire for each FrameBurstStart trigger.

This feature is used only if the FrameBurstStart trigger is enabled and the FrameBurstEnd trigger is disabled. Note that the total number of frames captured is also conditioned by AcquisitionFrameCount if AcquisitionMode is MultiFrame and ignored if AcquisitionMode is Single.

5.4.9 AcquisitionFrameRate

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

Controls the acquisition rate (in Hertz) at which the frames are captured.

TriggerMode must be **Off** for the Frame trigger.

5.4.10 AcquisitionLineRate

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

Controls the rate (in Hertz) at which the Lines in a Frame are captured.

TriggerMode must be **Off** for the Line trigger.

This is generally useful for line scan camera only.

5.4.11 AcquisitionStatusSelector

Name	AcquisitionStatusSelector
Category	AcquisitionControl
Level	Recommended
Interface	IEnumeration

Access	Read/Write
Unit	-
Visibility	Expert
Values	AcquisitionTriggerWait AcquisitionActive AcquisitionTransfer FrameTriggerWait FrameActive ExposureActive FrameTransfer (Deprecated)

Selects the internal acquisition signal to read using AcquisitionStatus**.**

See

Figure 5-1 and Figure 5-3 for details.Possible values are:

- **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 start trigger.
- **FrameActive**: Device is currently doing the capture of a frame.
- **ExposureActive**: Device is doing the exposure of a frame.
- **FrameTransfer (Deprecated)**: See TransferStatus.

5.4.12 AcquisitionStatus

Name	AcquisitionStatus[AcquisitionStatusSelector]
Category	AcquisitionControl
Level	Recommended
Interface	IBoolean
Access	Read
Unit	-
Visibility	Expert
Values	True False

Reads the state of the internal acquisition signal selected using **AcquisitionStatusSelector.**

5.5 Trigger Control features

The Trigger Control 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** (

Figure 5-1)**,** of a **Burst** of **Frames** (Figure 5-2), of individual **Frames** (Figure 5-3) or of each **Line** of a Frame for a line scan device (Figure 5-4)**. Triggers** 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, FrameBurstStart, FrameBurstEnd, FrameBurstActive, FrameStart, FrameEnd, FrameActive, LineStart, ExposureStart, ExposureEnd** and **ExposureActive.**

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

TriggerSource specifies the physical input **Line** or internal signal to use for the selected trigger. Standard trigger sources are: **Software, Line0, Line1, …, Counter0Start, Counter0End, …, Timer0Start, Timer0End, , …, UserOutput0, UserOutput1, …, Action0, Action1, …**

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.

TriggerDelay specifies the delay to apply after the trigger signal reception before effectively activating it.

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

Note that, a trigger is considered valid after the Dividers, Multipliers, Delay, ...

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 the 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.

The drawing below shows the functional model of the trigger generation in SFNC.

It shows the order and the stages that an input signal received on an external line may pass to become a valid trigger (Ex: a FrameTrigger).

Note that the signal received on an external line or generated from an internal circuit is considered a valid trigger for the Acquisition section only after passing all those stages (if they re implemented).

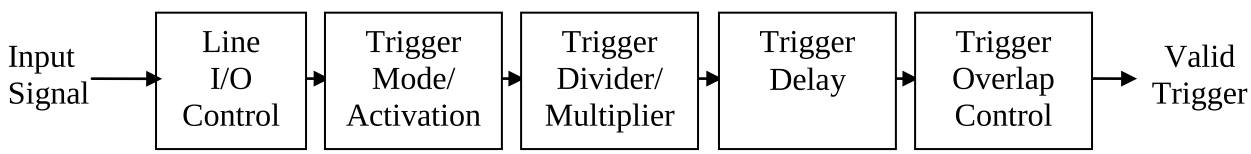


Figure 5-16: Trigger generation functional model.

- **Input Signal** represents an electrical signal received on an external Line or an internal signal.
- **Line I/O Control** represents the Line Control Block as described in the Figure 6-1.
- **Trigger Mode/Activation** represents the effect of the TriggerMode and TriggerActivation features.
- **Trigger Divider/Multiplier** represents the effect of the TriggerDivider and TriggerMultiplier features.
- **Trigger Delay** represents the effect of the TriggerDelay feature
- **Trigger Overlap Control** represents the effect of the TriggerOverlap feature.
- **Valid Trigger** represent when an incoming signal becomes a valid Trigger.

5.5.1 TriggerSelector

Name	TriggerSelector
Category	AcquisitionControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	AcquisitionStart

	AcquisitionEnd AcquisitionActive FrameStart FrameEnd FrameActive FrameBurstStart FrameBurstEnd FrameBurstActive LineStart ExposureStart ExposureEnd ExposureActive

Selects the type of trigger to configure.

See

Figure 5-1 and Figure 5-3 for details.

Possible values are:

- **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. The Acquisition is activated when the trigger signal becomes active and terminated when it goes back to the inactive state.
- **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).
- **FrameBurstStart**: Selects a trigger starting the capture of the bursts of frames in an acquisition. **AcquisitionBurstFrameCount** controls the length of each burst unless a FrameBurstEnd trigger is active. The total number of frames captured is also conditioned by AcquisitionFrameCount if AcquisitionMode is MultiFrame.
- **FrameBurstEnd**: Selects a trigger ending the capture of the bursts of frames in an acquisition.
- **FrameBurstActive**: Selects a trigger controlling the duration of the capture of the bursts of frames in an acquisition.

- **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).

5.5.2 TriggerMode

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

Controls if the selected trigger is active.

Possible values are:

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

5.5.3 TriggerSoftware

Name	TriggerSoftware[TriggerSelector]
Category	AcquisitionControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Beginner

Values	-

Generates an internal trigger. **TriggerSource** must be set to **Software.**

5.5.4 TriggerSource

Name	TriggerSource[TriggerSelector]
Category	AcquisitionControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Software SoftwareSignal0 (If 0 based), SoftwareSignal1, SoftwareSignal2, … Line0 (If 0 based), Line1, Line2, … Counter0Start (If 0 based), Counter1Start, Counter2Start, … Counter0End (If 0 based), Counter1End, Counter2End, … Timer0Start (If 0 based), Timer1Start, Timer2Start, … Timer0End (If 0 based), Timer1End, Timer2End, … UserOutput0, UserOutput1, UserOutput2, ... Action0 (If 0 based), Action1, Action2, … CC1, CC2, CC3, CC4, …

Specifies the internal signal or physical input **Line** to use as the trigger source. The selected trigger must have its **TriggerMode** set to **On**.

Possible values are:

- **Software**: Specifies that the trigger source will be generated by software using the **TriggerSoftware** command.

- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2, …**: Specifies that the trigger source will be a signal generated by software using the **SoftwareSignalPulse** 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.
- **Counter0Start, Counter1Start, Counter2Start, …, Counter0End, Counter1End, Counter2End, …**: Specifies which of the Counter signal to use as internal source for the trigger.
- **Timer0Start, Timer1Start, Timer2Start, …, Timer0End, Timer1End, Timer2End, …**: Specifies which Timer 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.
- **Action0, Action1, Action2, …**: Specifies which Action command to use as internal source for the trigger.
- **CC1, CC2, CC3, CC4**: Index of the Camera Link physical line and associated I/O control block to use. This ensures a direct mapping between the lines on the frame grabber and on the camera. Applicable to CameraLink products only.

5.5.5 TriggerActivation

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

Specifies the activation mode of the trigger.

Possible values are:

- **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.

5.5.6 TriggerOverlap

Name	TriggerOverlap[TriggerSelector]
Category	AcquisitionControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off ReadOut PreviousFrame

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.

Possible values are:

- **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.

5.5.7 TriggerDelay

Name	TriggerDelay[TriggerSelector]
Category	AcquisitionControl
Level	Recommended
Interface	IFloat

Access	Read/Write
Unit	us
Visibility	Expert
Values	Device-specific

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

5.5.8 TriggerDivider

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

Specifies a division factor for the incoming trigger pulses.

5.5.9 TriggerMultiplier

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

Specifies a multiplication factor for the incoming trigger pulses. It is used generally used in conjunction with **TriggerDivider** to control the ratio of triggers that are accepted.

5.6 Exposure Control features

The Exposure Control 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 programming the duration using the **ExposureTime or ExposureAuto** features.

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

```
Camera.ExposureMode = Timed;
Camera.ExposureTime = 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;
```

5.6.1 ExposureMode

Name	ExposureMode
Category	AcquisitionControl
Level	Recommended

Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Off Timed TriggerWidth TriggerControlled

Sets the operation mode of the Exposure (or shutter).

Possible values are:

- **Off**: Disables the Exposure and let the shutter open.
- **Timed**: Timed exposure. The exposure duration time is set using the ExposureTime or ExposureAuto features and the exposure starts with the FrameStart or LineStart (see Figure 5-3 and Figure 5-4).
- **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;

ExposureTime = 200;
```

Then the Exposure will be controlled using the **ExposureTime** 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.

5.6.2 ExposureTime

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

Sets the Exposure time when **ExposureMode** is **Timed** and ExposureAuto is Off. This controls the duration where the photosensitive cells are exposed to light.

5.6.3 ExposureAuto

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

Sets the automatic exposure mode 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.

Possible values are:

- **Off**: Exposure duration is user controlled using **ExposureTime.**
- **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 Control

The Digital I/O chapter covers the features required to control the general Input and Output signals of the device. This includes input and output control signals for Triggers Timers, counters and also static signals such as user configurable input or output bits.

The Digital I/O Control model presents 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.

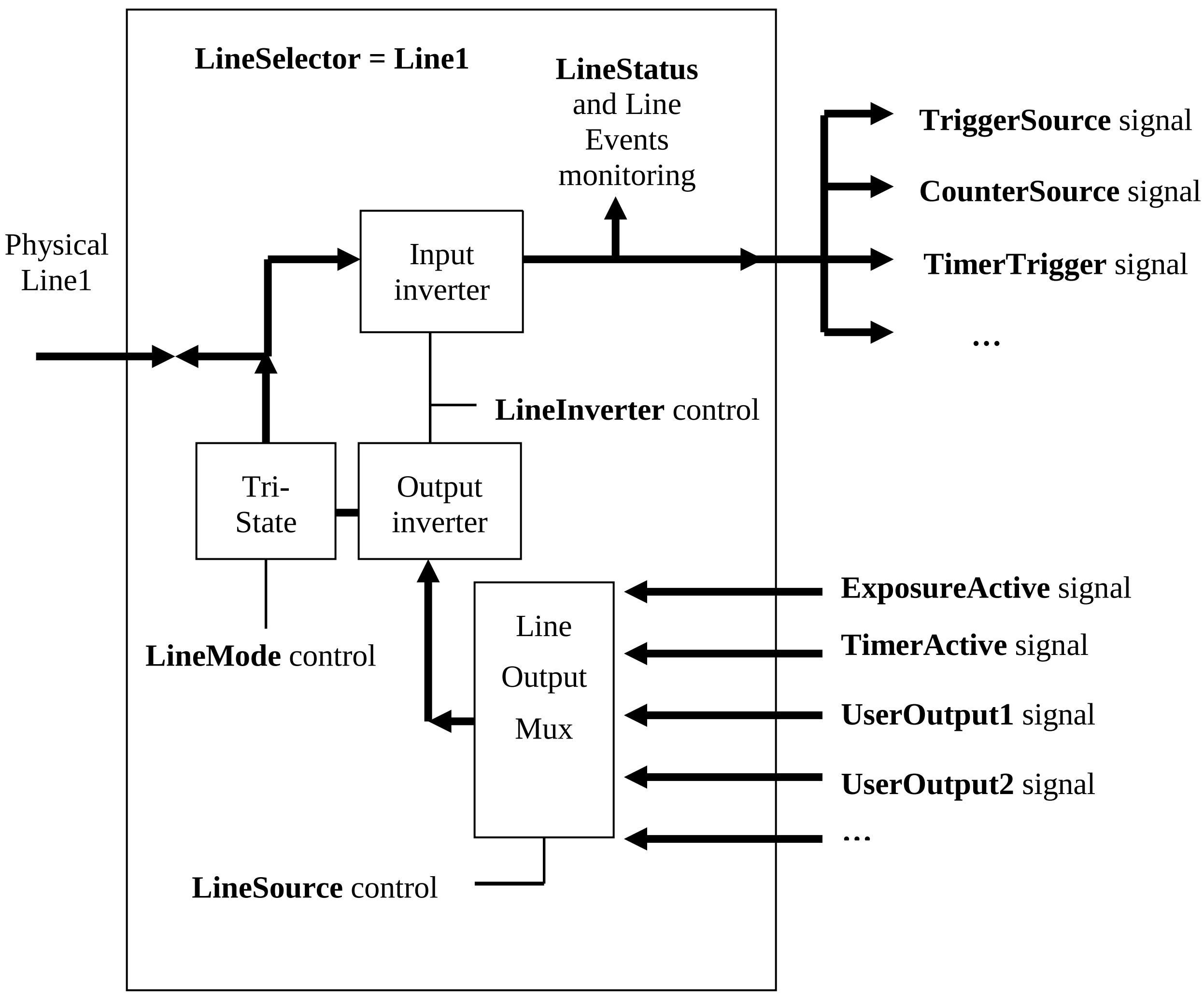


Figure 6-1: I/O Control

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 device connector, use the following code:

```
LineSelector = Line2;
LineMode     = Output;
LineInverter = True;
LineSource   = Timer1Active;
```

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.

See the **I/O Examples** section for more complete use cases of the I/O usage.

6.1 DigitalIOControl

Name	DigitalIOControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains the digital input and output control features.

6.2 LineSelector

Name	LineSelector
Category	DigitalIOControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Line0 (If 0 based), Line1, Line2, … *CC1, CC2, CC3, CC4,* …

Selects the 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.

Possible values are:

- **Line0** (If 0 based)**, Line1, Line2, ...**: Index of the physical line and associated I/O control block to use.

- **CC1, CC2, CC3, CC4**: Index of the Camera Link physical line and associated I/O control block to use. This ensures a **direct** mapping between the lines on the frame grabber and on the camera. Applicable to CameraLink Product only.

6.3 LineMode

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

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.

Possible values are:

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

6.4 LineInverter

Name	LineInverter[LineSelector]
Category	DigitalIOControl
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Expert
Values	False True

Controls the inversion of the signal of the selected input or output Line.

Possible values are:

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

6.5 LineStatus

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

Returns 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.

Possible values are:

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

6.6 LineStatusAll

Name	LineStatusAll
Category	DigitalIOControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	Device-specific

Returns the current status of all available Line signals at time of polling in a single bitfield.

The order is Line0 (If 0 based), Line1, Line2,...

6.7 LineSource

Name	LineSource[LineSelector]
Category	DigitalIOControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off AcquisitionTriggerWait AcquisitionActive FrameTriggerWait FrameActive ExposureActive Stream0TransferActive, Stream1TransferActive, … Stream0TransferPaused, Stream1TransferPaused, … Stream0TransferStopping, Stream1TransferStopping, … Stream0TransferStopped, Stream1TransferStopped, … Stream0TransferOverflow Stream1TransferOverflow, ... Counter0Active, Counter1Active, Counter2Active, … Timer0Active, Timer1Active, Timer2Active, … UserOutput0, UserOutput1, UserOutput2, …

Selects which internal acquisition or I/O source signal to output on the selected Line. **LineMode** must be **Output.**

See

Figure 6-1 for details.

Possible values are:

- **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 start trigger.
- **FrameActive**: Device is currently doing the capture of a Frame.
- **ExposureActive**: Device is doing the exposure of a Frame (or Line).
- **Counter0Active, Counter1Active, Counter2Active,** …: The chosen counter is in active state (counting).
- **Stream0TransferActive, Stream1TransferActive,** …: Transfer on the stream is active.
- **Stream0TransferPaused, Stream1TransferPaused,** …: Transfer on the stream is paused.
- **Stream0TransferStopping, Stream1TransferStopping,** …: Transfer on the stream is stopping.
- **Stream0TransferStopped, Stream1TransferStopped,** …: Transfer on the stream is stopped.
- **Stream0TransferOverflow, Stream1TransferOverflow,** …: Transfer on the stream is in overflow.
- **Timer0Active, Timer1Active, Timer2Active,** …: The chosen Timer is in active state.
- **UserOutput0, UserOutput1, UserOutput2,** …: The chosen User Output Bit state as defined by its current **UserOutputValue**.

6.8 LineFormat

Name	LineFormat[LineSelector]
Category	DigitalIOControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	NoConnect TriState

	TTL LVDS RS422 OptoCoupled

Controls the current electrical format of the selected physical input or output **Line**.

Possible values are:

- **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.

6.9 UserOutputSelector

Name	UserOutputSelector
Category	DigitalIOControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	UserOutput0, UserOutput1, UserOutput2, …

Selects which bit of the User Output register will be set by **UserOutputValue**.

Possible values are:

- **UserOutput0:** Selects the bit 0 of the User Output register.
- **UserOutput1**: Selects the bit 1 of the User Output register.
- **UserOutput2**: Selects the bit 2 of the User Output register.
- …

6.10 UserOutputValue

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

Sets the value of the bit selected by UserOutputSelector.

6.11 UserOutputValueAll

Name	UserOutputValueAll
Category	DigitalIOControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	Device-specific

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 (If 0 based). This means that the least significant bit of **UserOutputValueAll** corresponds to the UserOutput0 (if 0 based).

6.12 UserOutputValueAllMask

Name	UserOutputValueAllMask
Category	DigitalIOControl

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

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 Counter and Timer Control

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

A Counter is used to count internal events (FrameStart, Timer1End, …), I/O external events (Input Line rising edge, …) and even clock ticks. It can be reset, read or written 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 generate a 300 us strobe pulse coming from the Timer 1 when a rising edge trigger is detected on the physical Line 2 of the device connector, use the following code:

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

To set the destination output Line of the Timer pulse, see for example **Timer1Active** entry of 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.

See the Counter and Timer Examples section for more complete use cases of the counters and timers usage.

7.1 CounterAndTimerControl

Name	CounterAndTimerControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains the Counter and Timer control features.

7.2 CounterSelector

Name	CounterSelector
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Counter0 (If 0 based), Counter1, Counter2, …

Selects which Counter to configure.

Possible values are:

- **Counter0**: Selects the counter 0.
- **Counter1**: Selects the counter 1.
- **Counter2**: Selects the counter 2.
- …

7.3 CounterEventSource

Name	CounterEventSource[CounterSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart

	FrameEnd FrameBurstStart FrameBurstEnd LineStart LineEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2, … Counter0Start (If 0 based), Counter1Start, Counter2Start, … Counter0End (If 0 based), Counter1End, Counter2End, … Timer0Start (If 0 based), Timer1Start, Timer2Start, … Timer0End (If 0 based), Timer1End, Timer2End, … TimestampTick, SoftwareSignal0 (If 0 based), SoftwareSignal1, SoftwareSignal2, … Action0 (If 0 based), Action1, Action2, …

Select the events that will be the source to increment the Counter.

See

Figure 5-1, Figure 5-3 and

Figure 5-4 for details.

Possible values are:

- **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 Start Trigger.
- **FrameStart**: Counts the number of Frame Start.
- **FrameEnd**: Counts the number of Frame End.
- **FrameBurstStart**: Counts the number of Frame Burst Start.
- **FrameBurstEnd**: Counts the number of Frame Burst 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.
- **Line0, Line1, Line2,** …: Counts the number of transitions on the chosen I/O Line.
- **Counter0Start, Counter1Start, Counter2Start,** …: Counts the number of Counter Start.
- **Counter0End, Counter1End, Counter2End,** …: Counts the number of Counter End.
- **Timer0Start, Timer1Start, Timer2Start,** …: Counts the number of Timer Start pulses generated.
- **Timer0End, Timer1End, Timer2End,** …: Counts the number of Timer End pulses generated.
- **TimestampTick**: Counts the number of clock ticks of the Timestamp clock. Can be used to create a programmable timer.
- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2**, …: Counts the number of Software Signal.
- **Action0, Action1, Action2, …** : Counts the number of assertions of the chosen action signal.

7.4 CounterEventActivation

Name	CounterEventActivation[CounterSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge

Selects the Activation mode Event Source signal.

Possible values are:

- **RisingEdge**: Counts on the Rising Edge of the signal.

- **FallingEdge**: Counts on the Falling Edge of the signal.
- **AnyEdge**: Counts on the Falling or rising Edge of the selected signal.

7.5 CounterResetSource

Name	CounterResetSource[CounterSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off CounterTrigger AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2, … Counter0Start (If 0 based), Counter1Start, Counter2Start, … Counter0End (If 0 based), Counter1End, Counter2End, … Timer0Start (If 0 based), Timer1Start, Timer2Start, … Timer0End (If 0 based), Timer1End, Timer2End, … UserOutput0, UserOutput1, UserOutput2,… SoftwareSignal0 (If 0 based), SoftwareSignal1, SoftwareSignal2, … Action0 (If 0 based), Action1, Action2, …

Selects the signals that will be the source to reset the Counter.

Possible values are:

- **Off**: Disable the Counter Reset trigger.
- **CounterTrigger**: Resets with the reception of a trigger on the **CounterTriggerSource**.
- **AcquisitionTrigger**: Resets with the reception of the Acquisition Trigger.
- **AcquisitionStart**: Resets with the reception of the Acquisition Start.
- **AcquisitionEnd**: Resets with the reception of the Acquisition End.
- **FrameTrigger**: Resets with the reception of the Frame Start Trigger.
- **FrameStart**: Resets with the reception of the Frame Start.
- **FrameEnd**: Resets with the reception of the Frame End.
- **FrameBurstStart**: Resets with the reception of the Frame Burst Start.
- **FrameBurstEnd**: Resets with the reception of the Frame Burst End.
- **ExposureStart**: Resets with the reception of the Exposure Start.
- **ExposureEnd**: Resets with the reception of the Exposure End.
- **Line0** (If 0 based)**, Line1, Line2,** …: Resets by the chosen I/O Line.
- **UserOutput0, UserOutput1, UserOutput2,** …: Resets by the chosen User Output bit.
- **Counter0Start, Counter1Start, Counter2Start,** …: Resets with the reception of the Counter Start.
- **Counter0End, Counter1End, Counter2End,** …: Resets with the reception of the Counter End.
- **Timer0Start, Timer1Start, Timer2Start,** …: Resets with the reception of the Timer Start.
- **Timer0End, Timer1End, Timer2End,** …: Resets with the reception of the Timer End.
- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2**, …: Resets on the reception of the Software Signal.
- **Action0, Action1, Action2**, …: Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).

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

7.6 CounterResetActivation

Name	CounterResetActivation[CounterSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration

Access	Read/Write
Unit	-
Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Selects the Activation mode of the Counter Reset Source signal.

Possible values are:

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

7.7 CounterReset

Name	CounterReset[CounterSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Does a software reset of the selected Counter and starts it. The counter starts counting events immediately after the reset unless a Counter trigger is active. CounterReset can be used to reset the Counter independently from the CounterResetSource. 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.**

7.8 CounterValue

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

Reads or writes the current value of the selected Counter.

Writing to CounterValue is typically used to set the start value.

7.9 CounterValueAtReset

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

Reads the value of the selected Counter when it was reset by a trigger or by an explicit **CounterReset** command.

It represents the last counter value latched before reseting the counter.

7.10 CounterDuration

Name	CounterDuration[CounterSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IInteger

Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

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.**

7.11 CounterStatus

Name	CounterStatus[CounterSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Expert
Values	CounterIdle CounterTriggerWait CounterActive CounterCompleted CounterOverflow

Returns the current status of the Counter**.**

Possible values are:

- **CounterIdle**: The counter is idle.
- **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.

7.12 CounterTriggerSource

Name	CounterTriggerSource[CounterSelector]

Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd FrameBurstStart FrameBurstEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2, … UserOutput0, UserOutput1, UserOutput2,… Counter0Start (If 0 based), Counter1Start, Counter2Start, … Counter0End (if 0 based), Counter1End, Counter2End, … Timer0Start (If 0 based), Timer1Start, Timer2Start, … Timer0End (if 0 based), Timer1End, Timer2End, … SoftwareSignal0 (If 0 based), SoftwareSignal1, SoftwareSignal2, … Action0, Action1, Action2, …

Selects the source to start the Counter.

Possible values are:

- **Off**: Disables the Counter 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 Start Trigger.
- **FrameStart**: Starts with the reception of the Frame Start.
- **FrameEnd**: Starts with the reception of the Frame End.
- **FrameBurstStart**: Starts with the reception of the Frame Burst Start.
- **FrameBurstEnd**: Starts with the reception of the Frame Burst End.
- **ExposureStart**: Starts with the reception of the Exposure Start.
- **ExposureEnd**: Starts with the reception of the Exposure End.
- **Line0** (If 0 based)**, Line1, Line2, …**: Starts when the specified CounterTriggerActivation condition is met on the chosen I/O Line.
- **UserOutput0, UserOutput1, UserOutput2, …**: Specifies which User Output bit signal to use as internal source for the trigger.
- **Counter0Start, Counter1Start, Counter2Start, …**: Starts with the reception of the Counter Start.
- **Counter0End, Counter1End, Counter2End, …**: Starts with the reception of the Counter End.
- **Timer0Start, Timer1Start, Timer2Start, …:** Starts with the reception of the Timer Start.
- **Timer0End, Timer1End, Timer2End, …:** Starts with the reception of the Timer End.
- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2, …**: Starts on the reception of the Software Signal.
- **Action0, Action1, Action2, …: Starts** with the assertion of the chosen action signal.

7.13 CounterTriggerActivation

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

	LevelLow

Selects the activation mode of the trigger to start the Counter.

Possible values are:

- **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.

7.14TimerSelector

Name	TimerSelector
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Timer0 (if 0 based), Timer1, Timer2, …

Selects which Timer to configure.

Possible values are:

- **Timer0:** Selects the Timer 0.
- **Timer1:** Selects the Timer 1.
- **Timer2:** Selects the Timer 2.

7.15TimerDuration

Name	TimerDuration[TimerSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IFloat

Access	Read/Write
Unit	us
Visibility	Expert
Values	≥0

Sets the duration (in microseconds) of the Timer pulse.

When the Timer reaches the **TimerDuration** 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**.

7.16 TimerDelay

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

Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.

7.17 TimerReset

Name	TimerReset[TimerSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Does a software reset of the selected timer and starts it. The timer starts immediately after the reset unless a timer trigger is active.

7.18 TimerValue

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

Reads or writes the current value (in microseconds) of the selected Timer.

Writing **TimerValue** is typically used to set the start value.

7.19 TimerStatus

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

Returns the current status of the Timer**.**

Possible values are:

- **TimerIdle**: The Timer is idle.
- **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.

7.20 TimerTriggerSource

Name	TimerTriggerSource[TimerSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd FrameBurstStart FrameBurstEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2,… UserOutput0, UserOutput1, UserOutput2,… Counter0Start (If 0 based), Counter1Start, Counter2Start, … Counter0End (If 0 based), Counter1End, Counter2End, … Timer0Start (If 0 based), Timer1Start, Timer2Start, … Timer0End (If 0 based), Timer1End, Timer2End, … SoftwareSignal0 (If 0 based), SoftwareSignal1, SoftwareSignal2, … Action0 (If 0 based), Action1, Action2, …

Selects the source of the trigger to start the Timer.

Possible values are:

- **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 Start Trigger.
- **FrameStart**: Starts with the reception of the Frame Start.
- **FrameEnd**: Starts with the reception of the Frame End.
- **FrameBurstStart**: Starts with the reception of the Frame Burst Start.
- **FrameBurstEnd**: Starts with the reception of the Frame Burst End.
- **ExposureStart**: Starts with the reception of the Exposure Start.
- **ExposureEnd**: Starts with the reception of the Exposure End.
- **Line0** (If 0 based)**, Line1, Line2, …**: Starts when the specidfied TimerTriggerActivation condition is met on the chosen I/O Line.
- **UserOutput0, UserOutput1, UserOutput2, …**: Specifies which User Output bit signal to use as internal source for the trigger.
- **Counter0Start, Counter1Start, Counter2Start, …**: Starts with the reception of the Counter Start.
- **Counter0End, Counter1End, Counter2End, …**: Starts with the reception of the Counter End.
- **Timer0Start, Timer1Start, Timer2Start, …**: Starts with the reception of the Timer Start.
- **Timer0End, Timer1End, Timer2End, …**: Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.
- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2, …**: Starts on the reception of the Software Signal.
- **Action0, Action1, Action2, …:** Starts with the assertion of the chosen action signal.

7.21 TimerTriggerActivation

Name	TimerTriggerActivation[TimerSelector]
Category	CounterAndTimerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert

Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Selects the activation mode of the trigger to start the Timer.

Possible values are:

- **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 Event Control

This chapter 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 of the CCD exposure end to move the inspected part on a conveyer belt.

EventSelector selects which particular Event to control. There are many sources of events such as Acquisition, Timer, Counter and I/O lines.

The standard Acquisition related Events are: **AcquisitionTrigger, AcquisitionStart, AcquisitionEnd, AcquisitionTransferStart, AcquisitionTransferEnd, AcquisitionError, FrameTrigger, FrameStart, FrameEnd, FrameBurstStart, FrameBurstEnd, FrameTransferStart, FrameTransferEnd, ExposureStart, ExposureEnd** (See Figure 5-1 to Figure 5-4).

The standard Counters and Timers related Events are: **Counter0Start, Counter0End, Counter1Start, Counter1End, … Timer0Start, Timer0End, Timer1Start, Timer1End,** …

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 enable or disable the notification of the occurrence of the internal event selected by **EventSelector**. If **EventNotification** is **Off,** no event of the selected type is generated.

For each of the events listed in the **EventSelector** enumeration, there must be a corresponding feature with a standard name (ex: **EventExposureEnd**). The controlling application can rely on this event identifier to register a callback function to be notified that the event happened.

Also for each Event in **EventSelector**, there should be one category grouping all the related data members (Ex: **EventExposureEndData**).

The other data members in that category should also follow the naming convention described below (Ex: **EventExposureEndTimestamp**).

The recommended optional data members are:

- Timestamp: Unique timestamp of the Event.
- FrameID: Unique ID of the Frame (or image) that generated the Event.
- Followed by any other data related to this particular event.

Therefore, the naming convention for the Event related features is:

For each Event member of the **EventSelector** (Ex: **ExposureEnd**):

- You should provide an ICategory named:
Event prefix + "**EventName**" + **Data** postfix (Ex: **EventExposureEndData**)

- You must provide an IInteger Event feature that will be used to register the callback and is named:
Event prefix +"**EventName**" (Ex: **EventExposureEnd**).

- You should provide for each optional data member a corresponding feature named:
Event prefix + "**EventName**"+"**DataMember**" (Ex: **EventExposureEndTimestamp**).

For the **ExposureEnd** member of **EventSelector,** this would give:

ICategory **EventExposureEndData**

  IInteger **EventExposureEnd**

  IInteger **EventExposureEndTimeStamp**

  IInteger **EventExposureEndFrameID**

  …

With the above naming convention, for each Event listed in **EventSelector:**

  - A user always knows the name of the Feature to use to register a call back on that Event.

  - The user can take the parent of this feature to find the corresponding Event category.

  - In this Event category, the user will find all the features related to this Event.

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

```
Register(Camera.EventExposureEnd, CallbackDataObject, CallbackFunctionPtr)

Camera.EventSelector     = ExposureEnd;
Camera.EventNotification = On;
Camera.AcquisitionMode   = Continuous;
Camera.AcquisitionStart();
…
// In the callback of the ExposureEnd event, gets the event timestamp:
Timestamp = Camera.EventExposureEndTimestamp;
…
Camera.AcquisitionStop();
```

Here below, in addition to **EventControl**, **EventSelector** and **EventNotification** should be listed all the categories and data related features for each Event listed in the **EventSelector** enumeration feature.

For simplicity, all the categories and their data members are not listed explicitly in that document but a precise naming convention for the categories and their member is provided above instead.

Below, the detailled features for the members of the **EventSelector** are only listed for 3 typically recommended events: **FrameTrigger, ExposureEnd** and **Error**.

All the other members of the **EventSelector** feature should follow the exact same pattern for their features naming and category if they are present in a device.

See the **Acquisition and Trigger Examples** section for more complete use cases of the Events usage.

8.1 EventControl

Name	EventControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-

Visibility	Expert
Values	-

Category that contains Event control features.

8.2 EventSelector

Name	EventSelector
Category	EventControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	AcquisitionTrigger AcquisitionStart AcquisitionEnd AcquisitionTransferStart AcquisitionTransferEnd AcquisitionError FrameTrigger FrameStart FrameEnd FrameBurstStart FrameBurstEnd FrameTransferStart FrameTransferEnd ExposureStart ExposureEnd Stream0TransferStart Stream0TransferEnd Stream0TransferPause Stream0TransferResume Stream0TransferBlockStart Stream0TransferBlockEnd Stream0TransferBlockTrigger Stream0TransferBurstStart Stream0TransferBurstEnd Stream0TransferOverflow SequencerSetChange Counter0Start (If 0 based), Counter1Start, …

	Counter0End (If 0 based), Counter1End, … Timer0Start (If 0 based), Timer1Start, … Timer0End (If 0 based), Timer1End, … Line0RisingEdge (If 0 based), Line1RisingEdge, … Line0FallingEdge (If 0 based), Line1FallingEdge, … Line0AnyEdge (If 0 based), Line1AnyEdge, … ActionLate LinkSpeedChange Error Device-specific - GigE Vision Specific: PrimaryApplicationSwitch

Selects which Event to signal to the host application.

See

Figure 5-1 to

Figure 5-4 and

Figure 6-1 for details.

Possible values are:

- **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 to start the capture of one Frame.
- **FrameStart**: Device just started the capture of one Frame.
- **FrameEnd**: Device just completed the capture of one Frame.

- **FrameBurstStart**: Device just started the capture of a burst of Frames.
- **FrameBurstEnd**: Device just completed the capture of a burst of Frames.
- **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).
- **Stream0TransferStart**: Device just started the transfer of one or many Blocks.
- **Stream0TransferEnd**: Device just completed the transfer of one or many Blocks.
- **Stream0TransferPause**: Device just paused the transfer.
- **Stream0TransferResume**: Device just resumed the transfer.
- **Stream0TransferBlockStart**: Device just started the transfer of one Block.
- **Stream0TransferBlockEnd**: Device just completed the transfer of one Block.
- **Stream0TransferBlockTrigger**: Device just received a trigger to start the transfer of one Block.
- **Stream0TransferBurstStart**: Device just started the transfer of a burst of Blocks.
- **Stream0TransferBurstEnd**: Device just completed the transfer of a burst of Blocks.
- **Stream0TransferOverflow**: Device transfer queue overflowed.
- **SequencerSetChange**: Device sequencer set has chnged.
- **Counter0Start**: The event will be generated when counter 0 starts counting.
- **Counter0End**: The event will be generated when counter 0 ends counting.
- **Counter1Start**: The event will be generated when counter 1 starts counting.
- **Counter1End**: The event will be generated when counter 1 ends counting.
- **Timer0Start**: The event will be generated when Timer 0 starts counting.
- **Timer0End**: The event will be generated when Timer 0 ends counting.
- **Timer1Start**: The event will be generated when Timer 1 starts counting.
- **Timer1End**: The event will be generated when Timer 1 ends counting.
- **Line0RisingEdge**: The event will be generated when a Rising Edge is detected on the Line 0.
- **Line1RisingEdge**: The event will be generated when a Rising Edge is detected on the Line 1.

- **Line0FallingEdge**: The event will be generated when a Falling Edge is detected on the Line 0.
- **Line1FallingEdge**: The event will be generated when a Falling Edge is detected on the Line 1.
- **Line0AnyEdge**: The event will be generated when a Falling or Rising Edge is detected on the Line 0
- **Line1AnyEdge**: The event will be generated when a Falling or Rising Edge is detected on the Line 1.
- **ActionLate**: Then event will be generated when a valid scheduled action command is received and is scheduled to be executed at a time that is already past.
- **LinkSpeedChange**: Then event will be generated when the link speed has changed.
- **PrimaryApplicationSwitch**: Then event will be generated when a primary application switchover has been granted.
- **Error:** The event will be generated when the device encounter an error.
- …

8.3 EventNotification

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

Activate or deactivate the notification to the host application of the occurrence of the selected Event.

Possible values are:

- **Off**: The selected Event notification is disabled.
- **On**: The selected Event notification is enabled.

8.4 Frame Trigger Event (Example #1)

Below are the recommended features for the Frame Trigger Event handling.

8.4.1 EventFrameTriggerData

Name	EventFrameTriggerData
Category	EventControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains all the data features related to the FrameTrigger Event.

8.4.2 EventFrameTrigger

Name	EventFrameTrigger
Category	EventFrameTriggerData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the unique Identifier of the FrameTrigger type of Event. It can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type event received.

8.4.3 EventFrameTriggerTimestamp

Name	EventFrameTriggerTimestamp
Category	EventFrameTriggerData

Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the Timestamp of the AcquisitionTrigger Event. It can be used to determine precisely when the event occured.

8.4.4 EventFrameTriggerFrameID

Name	EventFrameTriggerFrameID
Category	EventFrameTriggerData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

8.5 Exposure End Event (Example #2)

Below are the recommended features for the Exposure End Event handling.

8.5.1 EventExposureEndData

Name	EventExposureEndData
Category	EventControl
Level	Recommended
Interface	ICategory

Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains all the data features related to the ExposureEnd Event.

8.5.2 EventExposureEnd

Name	EventExposureEnd
Category	EventExposureEndData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the unique identifier of the ExposureEnd type of Event. This feature can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type of event that will be received.

8.5.3 EventExposureEndTimestamp

Name	EventExposureEndTimestamp
Category	EventExposureEndData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the Timestamp of the ExposureEnd Event. It can be used to determine precisely when the event occured.

8.5.4 EventExposureEndFrameID

Name	EventExposureEndFrameID
Category	EventExposureEndData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the unique Identifier of the Frame (or image) that generated the ExposureEnd Event.

8.6 Error Event (Example #3)

Below are the recommended features for the Error Event handling.

8.6.1 EventErrorData

Name	EventErrorData
Category	EventControl
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains all the data features related to the Error Event.

8.6.2 EventError

Name	EventError

Category	EventErrorData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the unique identifier of the Error type of Event. It can be used to register a callback function to be notified of the Error event occurrence. Its value uniquely identifies that the event received was an Error.

8.6.3 EventErrorTimestamp

Name	EventErrorTimestamp
Category	EventErrorData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the Timestamp of the Error Event. It can be used to determine when the event occured.

8.6.4 EventErrorFrameID

Name	EventErrorFrameID
Category	EventErrorData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-

Visibility	Expert
Values	-

If applicable, returns the unique Identifier of the Frame (or image) that generated the Error Event.

8.6.5 EventErrorCode

Name	EventErrorCode
Category	EventErrorData
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns an error code for the error(s) that happened.

9 Analog Control

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

The **Gain**, **BlackLevel** and **Gamma** features will transform the original pixel value Y to a new value Y` according to the following formula:

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

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:

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

Other color camera controls each color channel gain independently, in which case, the Red, Green and Blue **Gain** 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. User 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 AnalogControl

Name	AnalogControl
Category	Root
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category that contains the Analog control features.

9.2 GainSelector

Name	GainSelector
Category	AnalogControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
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, …

Selects which Gain is controlled by the various Gain features.

In general, there are 2 types of gain that can exist 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 All gain are intended to be a pre or post stage amplification across all channels or taps, rather than a convenient way to set all the channels or tap gains in a single step.

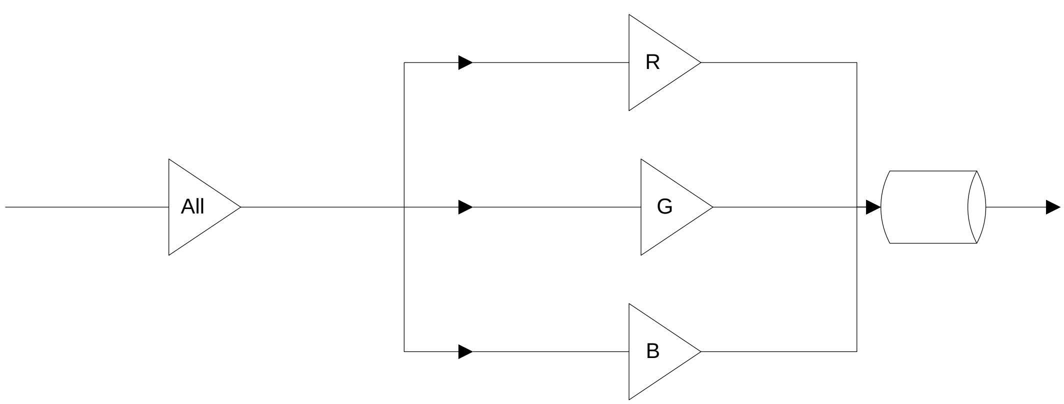


Figure 9-1: Gain All pre amplification

or

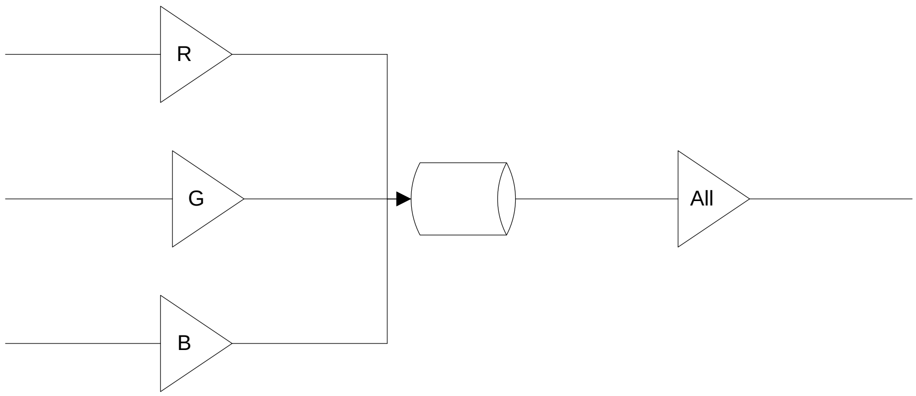


Figure 9-2: Gain All post amplification

By following this rule, the value read for the All gains remains valid even when the channel/tap gains are not all equal.

Possible values 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.
- …

9.3 Gain

Name	Gain[GainSelector]
Category	AnalogControl
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Device-specific

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 GenICam XML device description file.

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

9.4 GainAuto

Name	GainAuto[GainSelector]

Category	AnalogControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Off Once Continuous Device-specific

Sets the automatic gain control (AGC) mode. 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.

Possible values are:

- **Off**: Gain is User controlled using **Gain.**
- **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
Category	AnalogControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Off Once Continuous

	Device-specific

Sets the mode for automatic gain balancing between the sensor color channels or taps. The gain coefficients of each channel or tap are adjusted so they are matched.

Possible values are:

- **Off**: Gain tap balancing is user controlled using **Gain** .
- **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
Category	AnalogControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	All Red Green Blue Y U V Tap1, Tap2, …

Selects which Black Level is controlled by the various Black Level features.

The All Black Level selection is intended to be across all channels or taps, rather than a convenient way to set all the individual channels or taps black levels in a single step. By following this rule, the value read for the All black level remains valid even when the channel/tap black levels are not all equal.

Possible values 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 BlackLevel

Name	BlackLevel[BlackLevelSelector]
Category	AnalogControl
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Visibility	Expert
Values	Device-specific

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 GenICam XML device description file.

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

9.8 BlackLevelAuto

Name	BlackLevelAuto[BlackLevelSelector]
Category	AnalogControl
Level	Optional
Interface	IEnumeration
Access	Read/Write

Unit	-
Visibility	Expert
Values	Off Once Continuous Device-specific

Controls the mode for 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.

Possible values are:

- **Off**: Analog black level is user controlled using **BlackLevel**.
- **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.

9.9 BlackLevelAutoBalance

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

Controls the mode for automatic black level balancing between the sensor color channels or taps. The black level coefficients of each channel are adjusted so they are matched.

Possible values are:

- **Off**: Black level tap balancing is user controlled using **BlackLevel**.
- **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.

9.10 WhiteClipSelector

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

Selects which White Clip to control.

The **All** White Clip selection is intended to be across all channels or taps, rather than a convenient way to set all the individual channels or tap white clip in a single step. By following this rule, the value read for the All white clip remains valid even when the channel/tap white clips are not all equal.

Possible values 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.
- …

9.11 WhiteClip

Name	WhiteClip[WhiteClipSelector]
Category	AnalogControl
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Visibility	Expert
Values	Device-specific

Controls 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 GenICam XML device description file.

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

9.12 BalanceRatioSelector

Name	BalanceRatioSelector
Category	AnalogControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Red Green Blue

	Y U V Tap1, Tap2, …

Selects which Balance ratio to control.

Possible values are:

- **All**: White Clip will be applied to all channels or taps.
- **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.
- …

9.13 BalanceRatio

Name	BalanceRatio[BalanceRatioSelector]
Category	AnalogControl
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Visibility	Expert
Values	>0.0

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

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

C_w = **BalanceRatio** x C

where

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

BalanceRatio is the white balance coefficient.

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

9.14 BalanceWhiteAuto

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

Controls the mode for automatic white balancing between the color channels. The white balancing ratios are automatically adjusted.

Possible values are:

- **Off**: White balancing is user controlled using **BalanceRatioSelector** and **BalanceRatio**.
- **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.

9.15 Gamma

Name	Gamma
Category	AnalogControl

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

Controls the 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 Control

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

10.1 LUTControl

Name	LUTControl
Category	Root
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that includes the LUT control features.

10.2 LUTSelector

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

Selects which LUT to control.

Possible values are:

- **Luminance**: Selects the Luminace LUT.
- **Red**: Selects the Red LUT.

- **Green**: Selects the Green LUT.
- **Blue**: Selects the Blue LUT.

This feature is typically not available when only a single LUT is supported.

10.3 LUTEnable

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

Activates the selected LUT.

10.4 LUTIndex

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

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

10.5 LUTValue

Name	LUTValue[LUTSelector][LUTIndex]
Category	LUTControl
Level	Optional

Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru
Values	Device-specific

Returns the Value at entry **LUTIndex** of the LUT selected by **LUTSelector**.

10.6 LUTValueAll

Name	LUTValueAll[LUTSelector]
Category	LUTControl
Level	Optional
Interface	IRegister
Access	Read/Write
Unit	-
Visibility	Guru
Values	Device-specific

Accesses all the LUT coefficients in a single access without using individual **LUTIndex**.

11 GenICam Control

This chapter provides the necessary features to use the GenICam feature tree.

Note: In case of discrepancy between the features described in this chapter and the "GenICam Standard text" the SFNC document prevail.

11.1 Root

Name	Root
Category	None
Level	Mandatory
Interface	ICategory
Access	Read
Unit	-
Mandatory Visibility	Beginner
Values	-

Provides the Root of the GenICam features tree.

11.2 Device

Name	Device
Category	None
Level	Mandatory
Interface	IPort
Access	Read/Write
Unit	-
Visibility	Invisible
Values	-

Provides the default GenICam port of the Device.

Note: **Device** is the name of the standard port that is used to connect the node map to the transport layer and access the control port of the device. **Device** is a port node (not a feature node) and is generally not accessed by the end user directly. **Device** must not be included in the root feature tree.

11.3 TLParamsLocked

Name	TLParamsLocked
Category	None
Level	Mandatory
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Invisible
Values	-

Used by the Transport Layer to prevent critical features from changing during acquisition.

Possible values are:

- 0 : No features are locked.
- 1 : Transport Layer and Device critical features are locked and cannot be changed.

12 Transport Layer Control

This chapter provides the Transport Layer control features.

The TransportLayerControl category:

The generic features are under the TransportLayerControl category. Other Transport Layer specific features are under their respective sub category.

The GigEVision category:

This category lists the features necessary to access 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 discrepancy.

Note: If the user has configured the camera front end, he can read from the back end which **PayloadSize** will be transferred for each image. This number covers all kind of data coming with the image, e.g. timestamps etc. If the user allocates **PayloadSize** for each buffer he is insured that each frame will fit into his target buffers.

The CameraLink category:

This category lists the features necessary to access Camera Link devices and other information related to the camera Link transport medium.

12.1 Transport Layer features

This section describes the general transport layer features.

12.1.1 TransportLayerControl

Name	TransportLayerControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category that contains the transport Layer control features.

12.1.2 PayloadSize

Name	PayloadSize
Category	TransportLayerControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	B
Visibility	Expert
Values	≥0

Provides the number of bytes transferred for each image or chunk on the stream channel. This includes any end-of-line, end-of-frame statistics or other stamp data. This is the total size of data payload for a data block.

For the devices supporting multiple Stream Channels, the DeviceStreamChannelSelector feature should be used to select PayloadSize. This permits to inquire the payload size of each Stream channel individually.

This feature 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 timestamp data has typically a **PayloadSize** equals to (width x height x pixel size) in bytes. But it is strongly recommended to retrieve **PayloadSize** from the camera instead of relying on the above formula.

This feature is generaly mandatory for transmitters and transceivers of most Transport Layers.

12.1.3 DeviceTapGeometry

Name	DeviceTapGeometry
Category	TransportLayerControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Geometry_1X_1Y Geometry_1X2_1Y Geometry_2X_1Y Geometry_2XE_1Y Geometry_2XM_1Y Geometry_1X_1Y2 Geometry_1X_2YE Geometry_1X3_1Y Geometry_3X_1Y Geometry_1X Geometry_1X2 Geometry_2X Geometry_2XE Geometry_2XM Geometry_1X3 Geometry_3X Geometry_1X4_1Y Geometry_4X_1Y Geometry_2X2_1Y Geometry_2X2E_1Y Geometry_2X2M_1Y Geometry_1X2_2YE Geometry_2X_2YE Geometry_2XE_2YE Geometry_2XM_2YE Geometry_1X4

	Geometry_4X Geometry_2X2 Geometry_2X2E Geometry_2X2M Geometry_1X8_1Y Geometry_8X_1Y Geometry_4X2_1Y Geometry_2X2E_2YE Geometry_1X8 Geometry_8X Geometry_4X2 Geometry_4X2E Geometry_4X2E_1Y Geometry_1X10_1Y Geometry_10X_1Y Geometry_1X10 Geometry_10X …

This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

Possibles values are:

Geometry_1X_1Y: 1 X 1Y tap geometry.

Geometry_1X2_1Y: 1 X2 1Y tap geometry.

Geometry_2X_1Y: 2 X 1Y tap geometry.

Geometry_2XE_1Y: 2 XE 1Y tap geometry.

Geometry_2XM_1Y: 2 XM 1Y tap geometry.

Geometry_1X_1Y2: 1 X 1Y2 tap geometry.

Geometry_1X_2YE: 1 X 2YE tap geometry.

Geometry_1X3_1Y: 1 X3 1Y tap geometry.

Geometry_3X_1Y: 3 X 1Y tap geometry.

Geometry_1X: 1 X tap geometry.

Geometry_1X2: 1 X2 tap geometry.

Geometry_2X: 2 X tap geometry.

Geometry_2XE: 2 XE tap geometry.

Geometry_2XM: 2 XM tap geometry.

Geometry_1X3: 1 X3 tap geometry.

Geometry_3X: 3 X tap geometry.

Geometry_1X4_1Y: 1 X4 1Y tap geometry.

Geometry_4X_1Y: 4 X 1Y tap geometry.

Geometry_2X2_1Y: 2 X2 1Y tap geometry.

Geometry_2X2E_1Y: 2 X2E 1Y tap geometry.

Geometry_2X2M_1Y: 2 X2M 1Y tap geometry.

Geometry_1X2_2YE: 1 X2 2YE tap geometry.

Geometry_2X_2YE: 2 X 2YE tap geometry.

Geometry_2XE_2YE: 2 XE 2YE tap geometry.

Geometry_2XM_2YE: 2 XM 2YE tap geometry.

Geometry_1X4: 1 X4 tap geometry.

Geometry_4X: 4X tap geometry.

Geometry_2X2: 2 X2 tap geometry.

Geometry_2X2E: 2 X2E tap geometry.

Geometry_2X2M: 2 X2M tap geometry.

Geometry_1X8_1Y: 1 X8 1Y tap geometry.

Geometry_8X_1Y: X8 1Y tap geometry.

Geometry_4X2_1Y: 4 X2 1Y tap geometry.

Geometry_2X2E_2YE: 2 X2E 2YE tap geometry.

Geometry_1X8: 1 X8 tap geometry.

Geometry_8X: 8X tap geometry.

Geometry_4X2: 4 X2 tap geometry.

Geometry_4X2E: 4 X2E tap geometry.

Geometry_4X2E_1Y: 4 X2E 1Y tap geometry.

Geometry_1X10_1Y: 1 X10 1Y tap geometry.

Geometry_10X_1Y: 10X 1Y tap geometry.

Geometry_1X10: 1 X10 tap geometry.

Geometry_10X: X10 1Y tap geometry.

More detailed explanation, including graphical representation for every single Tap configuration is provided in the Tap Geometry Appendix of this document.

12.2 GigE Vision features

This section describes the GigE Vision specific transport layer features.

12.2.1 GigEVision

Name	GigEVision
Category	TransportLayerControl
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category that contains the features pertaining to the GigE Vision transport layer of the device.

12.2.2 GevVersionMajor (Deprecated)

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

This feature is deprecated (See DeviceTLVersionMajor). It was representing the major version of the specification.

For instance, GigE Vision version 1.0 would have the major version set to 1.

12.2.3 GevVersionMinor (Deprecated)

Name	GevVersionMinor
Category	GigEVision

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

This feature is deprecated (See DeviceTLVersionMinor). It was representing the minor version of the specification.

For instance, GigE Vision version 1.0 would have the minor version set to 0.

12.2.4 GevDeviceModelIsBigEndian (Deprecated)

Name	GevDeviceModelIsBigEndian
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Visibility	Guru
Values	True False

This feature is deprecated (See DeviceRegistersEndianess). It was representing the Endianess of the device registers.

12.2.5 GevDeviceClass (Deprecated)

Name	GevDeviceClass
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Guru

Values	Transmitter Receiver Transceiver Peripheral

This feature is deprecated (See DeviceType). It was representing the class of the device.

Note: The **GevDeviceClass** feature returns **Transmitter** for cameras.

12.2.6 GevDeviceModeCharacterSet (Deprecated)

Name	GevDeviceModeCharacterSet
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Guru
Values	UTF8

This feature is deprecated (See DeviceCharacterSet). It was representing the character set used by all the strings of the bootstrap registers.

12.2.7 GevPhysicalLinkConfiguration

Name	GevPhysicalLinkConfiguration
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	SingleLink MultiLink StaticLAG DynamicLAG

Controls the principal physical link configuration to use on next restart/power-up of the device.

12.2.8 GevCurrentPhysicalLinkConfiguration

Name	GevCurrentPhysicalLinkConfiguration
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Expert
Values	SingleLink MultiLink StaticLAG DynamicLAG

Indicates the current physical link configuration of the device.

Note: When Multi-Link and LAG configurations are used concurrently, the device shall report the LAG configuration.

12.2.9 GevActiveLinkCount (Deprecated)

Name	GevActiveLinkCount
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	> 0

Indicates the current number of active logical links.

12.2.10 GevSupportedOptionSelector

Name	GevSupportedOptionSelector
Category	GigEVision

Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	SingleLink MultiLink StaticLAG DynamicLAG PAUSEFrameReception PAUSEFrameGeneration IPConfigurationLLA IPConfigurationDHCP IPConfigurationPersistentIP StreamChannelSourceSocket StandardIDMode MessageChannelSourceSocket CommandsConcatenation WriteMem PacketResend Event EventData PendingAck IEEE1588 Action UnconditionalAction ScheduledAction PrimaryApplicationSwitchover ExtendedStatusCodes ExtendedStatusCodesVersion2_0 DiscoveryAckDelay DiscoveryAckDelayWritable TestData ManifestTable CCPApplicationSocket LinkSpeed HeartbeatDisable SerialNumber UserDefinedName StreamChannel0BigAndLittleEndian StreamChannel0IPReassembly StreamChannel0MultiZone

	StreamChannel0PacketResendDestination StreamChannel0AllInTransmission StreamChannel0UnconditionalStreaming StreamChannel0ExtendedChunkData StreamChannel1BigAndLittleEndian StreamChannel1IPReassembly StreamChannel1MultiZone StreamChannel1PacketResendDestination StreamChannel1AllInTransmission StreamChannel1UnconditionalStreaming StreamChannel1ExtendedChunkData StreamChannel2BigAndLittleEndian StreamChannel2IPReassembly StreamChannel2MultiZone StreamChannel2PacketResendDestination StreamChannel2AllInTransmission StreamChannel2UnconditionalStreaming StreamChannel2ExtendedChunkData …

Selects the GEV option to interrogate for existing support.

Note: The IP reassembly options (**StreamChannel0IPReassembly**, **StreamChannel1IPReassembly**, …) are only applicable to GVSP receiver stream channels.

12.2.11 GevSupportedOption

Name	GevSupportedOption[GevSupportedOptionSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read
Unit	-
Visibility	Expert
Values	True False

Returns if the selected GEV option is supported.

12.2.12 GevInterfaceSelector

Name	GevInterfaceSelector
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Beginner
Values	≥0

Selects which logical link to control.

Note: The number of physical network interfaces may be greater than the value reported by **GevInterfaceSelector**. This is generally the case when link aggregation is activated.

12.2.13 GevLinkSpeed (Deprecated)

Name	GevLinkSpeed[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	Mbs
Visibility	Expert
Values	>0

This feature is deprecated (See DeviceLinkSpeed). It was representing the speed of transmission negotiated by the given logical link.

12.2.14 GevMACAddress

Name	GevMACAddress[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IInteger

Access	Read
Unit	-
Visibility	Beginner
Values	≥0

MAC address of the logical link.

12.2.15 This feature must return a 64-bit value representing the full MAC address of the device i.e. the high and low parts. GevPAUSEFrameReception

Name	GevPAUSEFrameReception[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	True False

Controls whether incoming PAUSE Frames are handled on the given logical link.

12.2.16 GevPAUSEFrameTransmission

Name	GevPAUSEFrameTransmission[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	True False

Controls whether PAUSE Frames can be generated on the given logical link.

12.2.17 GevCurrentIPConfigurationLLA

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

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

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

12.2.18 GevCurrentIPConfigurationDHCP

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

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

12.2.19 GevCurrentIPConfigurationPersistentIP

Name	GevCurrentIPConfigurationPersistentIP[GevInterfaceSelector]
Category	GigEVision
Level	Optional

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

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

12.2.20 GevCurrentIPAddress

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

Reports the IP address for the given logical link.

12.2.21 GevCurrentSubnetMask

Name	GevCurrentSubnetMask[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Beginner
Values	≥0

Reports the subnet mask of the given logical link.

12.2.22 GevCurrentDefaultGateway

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

Reports the default gateway IP address to be used on the given logical link.

12.2.23 GevIPConfigurationStatus

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

Reports the current IP configuration status.

12.2.24 GevFirstURL (Deprecated)

Name	GevFirstURL
Category	GigEVision
Level	Optional
Interface	IString

Access	Read
Unit	-
Visibility	Guru
Values	-

Indicates the first URL to the GenICam XML device description file. The First URL is used as the first choice by the application to retrieve the GenICam XML device description file.

12.2.25 GevSecondURL(Deprecated)

Name	GevSecondURL
Category	GigEVision
Level	Optional
sequInterface	IString
Access	Read
Unit	-
Visibility	Guru
Values	-

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

12.2.26 GevNumberOfInterfaces (Deprecated)

Name	GevNumberOfInterfaces
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	>0

This feature is deprecated (See DeviceLinkSelector). It was representing the number of logical links supported by this device.

12.2.27 GevPersistentIPAddress

Name	GevPersistentIPAddress[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Beginner
Values	≥0

Controls the Persistent IP address for this logical link. It is only used when the device boots with the Persistent IP configuration scheme.

12.2.28 GevPersistentSubnetMask

Name	GevPersistentSubnetMask[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Beginner
Values	≥0

Controls the Persistent subnet mask associated with the Persistent IP address on this logical link. It is only used when the device boots with the Persistent IP configuration scheme.

12.2.29 GevPersistentDefaultGateway

Name	GevPersistentDefaultGateway[GevInterfaceSelector]
Category	GigEVision
Level	Optional
Interface	IInteger

Access	Read/Write
Unit	-
Visibility	Beginner
Values	≥0

Controls the persistent default gateway for this logical link. It is only used when the device boots with the Persistent IP configuration scheme.

12.2.30 GevMessageChannelCount (Deprecated)

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

This feature is deprecated (See DeviceEventChannelCount). It was representing the number of message channels supported by this device.

12.2.31 GevStreamChannelCount (Deprecated)

Name	GevStreamChannelCount
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	0 to 512

This feature is deprecated (See DeviceStreamChannelCount). It was representing the number of stream channels supported by this device.

12.2.32 GevHeartbeatTimeout (Deprecated)

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

This feature is deprecated (See DeviceLinkHeartbeatTimeout). It was controling the current heartbeat timeout in milliseconds.

12.2.33 GevTimestampTickFrequency (Deprecated)

Name	GevTimestampTickFrequency
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	Hz
Visibility	Expert
Values	≥0

This feature is deprecated (See the increment of the TimestampLatchValue feature). It was used to indicate the number of timestamp ticks in 1 second (frequency in Hz). If IEEE 1588 is used, this feature must return 1,000,000,000 (1 GHz).

This is a 64 bits number.

12.2.34 GevTimestampControlLatch (Deprecated)

Name	GevTimestampControlLatch
Category	GigEVision
Level	Optional
Interface	ICommand

Access	Write
Unit	-
Visibility	Expert
Values	-

This feature is deprecated (See TimestampLatch). It was used to latche the current timestamp counter into GevTimestampValue.

12.2.35 GevTimestampControlReset (Deprecated)

Name	GevTimestampControlReset
Category	GigEVision
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Visibility	Expert
Values	-

This feature is deprecated (See TimestampReset). It was used to reset the timestamp counter to 0. This feature is not available or as no effect when IEEE 1588 is used.

12.2.36 GevTimestampValue (Deprecated)

Name	GevTimestampValue
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	
Visibility	Expert
Values	≥ 0

This feature is deprecated (See TimestampLatchValue). It was used to return 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.

12.2.37 GevDiscoveryAckDelay

Name	GevDiscoveryAckDelay
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/(Write)
Unit	ms
Visibility	Expert
Values	≥0 and < 1000

Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

12.2.38 GevIEEE1588

Name	GevIEEE1588
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Expert
Values	True False

Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

12.2.39 GevIEEE1588ClockAccuracy

Name	GevIEEE1588ClockAccuracy
Category	GigEVision
Level	Optional
Interface	IEnumeration

Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Within25ns Within100ns Within250ns Within1us Within2p5u Within10us Within25us Within100us Within250us Within1ms Within2p5ms Within10ms Within25ms Within100ms Within250ms Within1s Within10s GreaterThan10s AlternatePTPProfile Unknown Reserved

Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

This feature maps to the IEEE 1588 clockAccuracy attribute.

12.2.40 GevIEEE1588Status

Name	GevIEEE1588Status
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Expert
Values	Initializing

	Faulty Disabled Listening PreMaster Master Passive Uncalibrated Slave

Provides the status of the IEEE 1588 clock.

12.2.41 GevGVCPExtendedStatusCodesSelector

Name	GevGVCPExtendedStatusCodesSelector
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	Version1_1 Version2_0

Selects the GigE Vision version to control extended status codes for.

12.2.42 GevGVCPExtendedStatusCodes

Name	GevGVCPExtendedStatusCodes[GevGVCPExtendedStatusCodesSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

Enables the generation of extended status codes.

12.2.43 GevGVCPPendingAck

Name	GevGVCPPendingAck
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

Enables the generation of PENDING_ACK.

12.2.44 GevGVCPHeartbeatDisable (Deprecated)

Name	GevGVCPHeartbeatDisable
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Expert
Values	True False

This feature is deprecated (See DeviceHeartbeatMode). It was used to disable the GVCP heartbeat.

12.2.45 GevGVCPPendingTimeout (Deprecated)

Name	GevGVCPPendingTimeout
Category	GigEVision
Level	Optional

Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

This feature is deprecated (See DeviceCommandTimeout). It was used to indicate the longest GVCP command execution time before a device returns a PENDING_ACK.

12.2.46 GevPrimaryApplicationSwitchoverKey

Name	GevPrimaryApplicationSwitchoverKey
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Write-Only
Unit	-
Visibility	Guru
Values	≥ 0

Controls the key to use to authenticate primary application switchover requests.

12.2.47 GevGVSPExtendedIDMode

Name	GevGVSPExtendedIDMode
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Off On

Enables the extended IDs mode.

If the standard IDs mode is not supported, this feature should return On and be read only.

This feature is not applicable for GigE Vision 1.x devices.

12.2.48 GevCCP

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

Controls the device access privilege of 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.

12.2.49 GevPrimaryApplicationSocket

Name	GevPrimaryApplicationSocket
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Returns the UDP source port of the primary application.

12.2.50 GevPrimaryApplicationIPAddress

Name	GevPrimaryApplicationIPAddress
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Returns the address of the primary application.

12.2.51 GevMCPHostPort

Name	GevMCPHostPort
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru
Values	≥0

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

12.2.52 GevMCDA

Name	GevMCDA
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-

Visibility	Guru
Values	≥0

Controls the destination IP address for the message channel.

12.2.53 GevMCTT

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

Provides the transmission timeout value in milliseconds.

12.2.54 GevMCRC

Name	GevMCRC
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru
Values	≥0

Controls the number of retransmissions allowed when a message channel message times out.

12.2.55 GevMCSP

Name	GevMCSP
Category	GigEVision
Level	Optional

Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

This feature indicates the source port for the message channel.

12.2.56 GevStreamChannelSelector

Name	GevStreamChannelSelector
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Selects the stream channel to control.

12.2.57 GevSCCFGPacketResendDestination

Name	GevSCCFGPacketResendDestination[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

Enables the alternate IP destination for stream packets resent due to a packet resend request. When True, the source IP address provided in the packet resend command packet is used. When False, the value set in the **GevSCDA[GevStreamChannelSelector]** feature is used.

This feature is only valid for GVSP transmitters.

12.2.58 GevSCCFGAllInTransmission

Name	GevSCCFGAllInTransmission[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

12.2.59 GevSCCFGUnconditionalStreaming

Name	GevSCCFGUnconditionalStreaming[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).

12.2.60 GevSCCFGExtendedChunkData

Name	GevSCCFGExtendedChunkData[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

Enables cameras to use the extended chunk data payload type for this stream channel.

12.2.61 GevSCPDirection (Deprecated)

Name	GevSCPDirection[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Guru
Values	Transmitter Receiver

This feature is deprecated (See DeviceStreamChannelType). It was used to report the direction of the stream channel.

12.2.62 GevSCPInterfaceIndex

Name	GevSCPInterfaceIndex[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read/(Write)

Unit	-
Visibility	Guru
Values	0 to 3

Index of the logical link to use.

Specific streams might be hard-coded to specific logical links. Therefore this field might be read-only on certain devices.

12.2.63 GevSCPHostPort

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

Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream. Setting this value to 0 closes the stream channel.

12.2.64 GevSCPSFireTestPacket

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

Sends a test packet. When this feature is set, the device will fire one test packet.

The "don`t fragment" bit of IP header must be set for this test packet.

12.2.65 GevSCPSDoNotFragment

Name	GevSCPSDoNotFragment[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

The state of this feature 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.

12.2.66 GevSCPSBigEndian (Deprecated)

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

This feature is deprecated (See DeviceStreamChannelEndianness). It was used to control the 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.

12.2.67 GevSCPSPacketSize (Deprecated)

Name	GevSCPSPacketSize[GevStreamChannelSelector]
Category	GigEVision
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	B
Visibility	Expert
Values	>0

This feature is deprecated (See DeviceStreamChannelPacketSize). It was used to specify the stream packet size, in bytes, to send on the selected channel for a GVSP transmitter or specifies the maximum packet size supported by a GVSP receiver.

This does not include 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).

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

12.2.68 GevSCPD

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

Controls the delay (in GEV 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 or the network infrastructure cannot keep up with the packets coming from the device.

12.2.69 GevSCDA

Name	GevSCDA[GevStreamChannelSelector]

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

Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

12.2.70 GevSCSP

Name	GevSCSP[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Indicates the source port of the stream channel.

12.2.71 GevSCZoneCount

Name	GevSCZoneCount[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru

Values	1 to 32

Reports the number of zones per block transmitted on the selected stream channel.

12.2.72 GevSCZoneDirectionAll

Name	GevSCZoneDirectionAll[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	00000000h to FFFFFFFFh

Reports the transmission direction of each zone transmitted on the selected stream channel.

This feature is represented as an unsigned integer. The most significant bit of the range of valid values reports the direction of the first zone (Zone ID 0) while the least significant bit represents the direction of the last zone (Zone ID 1).

12.2.73 GevSCZoneConfigurationLock

Name	GevSCZoneConfigurationLock[GevStreamChannelSelector]
Category	GigEVision
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Guru
Values	True False

Controls whether the selected stream channel multi-zone configuration is locked. When locked, the GVSP transmitter is not allowed to change the number of zones and their direction during block acquisition and transmission.

12.3 Network Statistics features

This section describes the network statistics specific features.

12.3.1 NetworkStatistics

Name	NetworkStatistics
Category	TransportLayerControl
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Guru
Values	-

Category that contains statistics pertaining to various modules of the GigE Vision transport layer.

12.3.2 oMACControlFunctionEntity

Name	oMACControlFunctionEntity
Category	NetworkStatistics
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Guru
Values	-

Category that contains statistics pertaining to the MAC control PAUSE function of the device.

When multiple links are aggregated together, this category represents the aggregated statistics of all associated MACs. This is because it is not possible to select each physical network interface in this case.

The counters in this section are defined by the IEEE 802.3 standard. They are generally nonresetable. Rollover behavior and maximum value are device-specific.

12.3.3 aPAUSEMACCtrlFramesTransmitted

Name	aPAUSEMACCtrlFramesTransmitted[GevInterfaceSelector]
Category	oMACControlFunctionEntity
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Reports the number of transmitted PAUSE frames.

12.3.4 aPAUSEMACCtrlFramesReceived

Name	aPAUSEMACCtrlFramesReceived[GevInterfaceSelector]
Category	oMACControlFunctionEntity
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Guru
Values	≥0

Reports the number of received PAUSE frames.

12.4 Camera Link features

This section describes the Camera Link specific features.

12.4.1 CameraLink

Name	CameraLink
Category	TransportLayerControl

Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category that contains the features pertaining to the Camera Link transport layer of the device.

This category is optional especially if the device supports only one transport layer.

12.4.2 ClConfiguration

Name	ClConfiguration
Category	CameraLink
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Beginner
Values	Base Medium Full DualBase EightyBit Deca (Deprecated)

This Camera Link specific feature describes the configuration used by the camera. It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitizationTaps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera.

Possible values are:

- **Base**: Standard base configuration described by the Camera Link standard.
- **Medium**: Standard medium configuration described by the Camera Link standard.
- **Full**: Standard full configuration described by the Camera Link standard.

- **DualBase**: The camera streams the data from multiple taps (that do not fit in the standard base configuration) through two Camera Link base ports. It is responsibility of the application or frame grabber to reconstruct the full image. Only one of the ports (fixed) serves as the "master" for serial communication and triggering.
- **EightyBit**: Standard 80-bit configuration with 10 taps of 8 bits or 8 taps of 10 bits, as described by the Camera Link standard.
- **Deca (Deprecated)**: This enumeration entry is deprecated. It was used for Deca configuration with 10 taps of 8-bit. Use EightyBit instead.

If the feature is omitted, one of the standard configurations (Base-Medium-Full) is expected. In that case the configuration can be unequivocally deduced from the SensorDigitizationTaps and PixelSize values.

12.4.3 ClTimeSlotsCount

Name	ClTimeSlotsCount
Category	CameraLink
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	One Two Three

This Camera Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Possible values are:

- **One**: One time slot.
- **Two**: Two time slots.
- **One**: Three time slots.

It indicates the number of consecutive time slots required to transfer one data of each tap.

12.5 CoaXPress features

This section describes the CoaXPress specific features.

12.5.1 CoaXPress

Name	CoaXPress
Category	TransportLayerControl
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category that contains the features pertaining to the CoaXPress transport layer of the device.

12.5.2 CxpLinkConfigurationStatus

Name	CxpLinkConfigurationStatus
Category	CoaXPress
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Beginner
Values	None Pending CXP1_X1 CXP2_X1 CXP3_X1 CXP5_X1 CXP6_X1 CXP1_X2 CXP2_X2 CXP3_X2 CXP5_X2 CXP6_X2 CXP1_X3 CXP2_X3 CXP3_X3 CXP5_X3 CXP6_X3

	CXP1_X4 CXP2_X4 CXP3_X4 CXP5_X4 CXP6_X4 CXP1_X5 CXP2_X5 CXP3_X5 CXP5_X5 CXP6_X5 CXP1_X6 CXP2_X6 CXP3_X6 CXP5_X6 CXP6_X6

This feature indicates the current and active Link configuration used by the Device.

When the Link is active, this feature returns the Link configuration as the combination of the Connection speed and the number of active Connections using the following format "CXP*m*_X*n*", where *m* is the Connection speed and *n* the number of active Connections. For example "CXP6_X4" means 4 connections are operating at CXP-6 speed (6.25 Gbps) so the total speed on the virtual single link is 25 Gbps.

Possible values are:

- **None**: The Link configuration of the Device is unknown. Either the configuration operation has failed or there is nothing connected.
- **Pending**: The Device is in the process of configuring the Link. The Link cannot be used yet.
- **CXP*1*_X*1***: 1 Connection operating at CXP-1 speed (1.25 Gbps).
- **CXP*2*_X*1***: 1 Connection operating at CXP-2 speed (2.50 Gbps).
- **CXP*3*_X*1***: 1 Connection operating at CXP-3 speed (3.125 Gbps).
- **CXP*5*_X*1***: 1 Connection operating at CXP-5 speed (5.00 Gbps).
- **CXP*6*_X*1***: 1 Connection operating at CXP-6 speed (6.25 Gbps).
- **CXP1_X2**: 2 Connections operating at CXP-1 speed (2.50 Gbps).
- …

12.5.3 CxpLinkConfigurationPreferred

Name	CxpLinkConfigurationPreferred

Category	CoaXPress
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Expert
Values	CXP1_X1 CXP2_X1 CXP3_X1 CXP5_X1 CXP6_X1 CXP1_X2 CXP2_X2 CXP3_X2 CXP5_X2 CXP6_X2 CXP1_X3 CXP2_X3 CXP3_X3 CXP5_X3 CXP6_X3 CXP1_X4 CXP2_X4 CXP3_X4 CXP5_X4 CXP6_X4 CXP1_X5 CXP2_X5 CXP3_X5 CXP5_X5 CXP6_X5 CXP1_X6 CXP2_X6 CXP3_X6 CXP5_X6 CXP6_X6

Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

The Link configuration is returned as the combination of the Connection speed and the number of active Connections using the following format "CXP*m*_X*n*", where *m* is the Connection speed and *n* the number of active Connections.

Possible values are:

- **CXP*1*_X*1***: 1 Connection operating at CXP-1 speed (1.25 Gbps).
- **CXP*2*_X*1***: 1 Connection operating at CXP-2 speed (2.50 Gbps).
- **CXP3_X*1***: 1 Connection operating at CXP-3 speed (3.125 Gbps).
- **CXP5_X*1***: 1 Connection operating at CXP-5 speed (5.00 Gbps).
- **CXP6_X*1***: 1 Connection operating at CXP-6 speed (6.25 Gbps).
- **CXP1_X2**: 2 Connections operating at CXP-1 speed (2.50 Gbps).
- …

12.5.4 CxpLinkConfiguration

Name	CxpLinkConfiguration
Category	CoaXPress
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Auto CXP1_X1 CXP2_X1 CXP3_X1 CXP5_X1 CXP6_X1 CXP1_X2 CXP2_X2 CXP3_X2 CXP5_X2 CXP6_X2 CXP1_X3 CXP2_X3 CXP3_X3 CXP5_X3 CXP6_X3 CXP1_X4 CXP2_X4

	CXP3_X4 CXP5_X4 CXP6_X4 CXP1_X5 CXP2_X5 CXP3_X5 CXP5_X5 CXP6_X5 CXP1_X6 CXP2_X6 CXP3_X6 CXP5_X6 CXP6_X6

This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device. In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus.

The Link configuration is specified as the combination of the Connection speed and the number of active Connections using the following format "CXP*m*_X*n*", where *m* is the Connection speed and *n* the number of active Connections. Selecting Auto sets the Link to normal, automatic, discovery, as described in the CoaXPress standard. The Receiver Device will automatically discover any Transmitter Device connected from then on.

Possible values are:

- **Auto**: Sets Automatic discovery for the Link Configuration.
- **CXP*1*_X*1***: Force the Link to 1 Connection operating at CXP-1 speed (1.25 Gbps).
- **CXP*2*_X*1***: Force the Link to 1 Connection operating at CXP-2 speed (2.50 Gbps).
- **CXP3_X*1***: Force the Link to 1 Connection operating at CXP-3 speed (3.125 Gbps).
- **CXP5_X*1***: Force the Link to 1 Connection operating at CXP-5 speed (5.00 Gbps).
- **CXP6_X*1***: Force the Link to 1 Connection operating at CXP-6 speed (6.25 Gbps).
- **CXP1_X2**: Force the Link to 2 Connections operating at CXP-1 speed (2.50 Gbps).
- …

12.5.5 CxpConnectionSelector

Name	CxpConnectionSelector

Category	CoaXPress
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Selects the CoaXPress physical connection to control.

Note that this selector should be set to 0, or omitted, when controlling features relating to the CoaXPress uplink from the Receiver Device to the Transmitter Device, because only one connection 0 is used for this purpose.

12.5.6 CxpConnectionTestMode

Name	CxpConnectionTestMode[CxpConnectionSelector]
Category	CoaXPress
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off Mode1

Enables the test mode for an individual physical connection of the Device.

Possible values are:

- **Off**: Test mode is disabled.
- **Mode1**: Test mode is one.

This can be used to test communication errors of the system cabling between devices.
When enabled, this feature results in special test packets being sent continuously by the Device on the connection specified by CxpConnectionSelector.

The Device receiving the test packet on the other end of the connection can check for errors by reading its own corresponding CxpConnectionTestErrorCount and CxpConnectionTestPacketCount features.

Typically, the test will need to be run for some time (e.g. minutes) to get a meaningful error rate.

12.5.7 CxpConnectionTestErrorCount

Name	CxpConnectionTestErrorCount[CxpConnectionSelector]
Category	CoaXPress
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

The transmission of those test packets is enabled by the CxpConnectionTestMode feature of the Device on the other end of the connection under test.

This feature can be read at any time while a test is running. It can be written to zero when a test is not running to reset the counter between tests.

12.5.8 CxpConnectionTestPacketCount

Name	CxpConnectionTestPacketCount[CxpConnectionSelector]
Category	CoaXPress
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Reports the current count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

The transmission of those test packets is enabled by the CxpConnectionTestMode feature of the Device on the other end of the connection under test.

This feature can be read at any time while a test is running. It can be written to zero when the test is not running to reset the counter between tests.

12.5.9 CxpPoCxpAuto

Name	CxpPoCxpAuto
Category	CoaXPress
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Visibility	Expert
Values	-

Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

This feature shall be present only on receiver or transceiver Devices controlling PoCXP.

12.5.10 CxpPoCxpTurnOff

Name	CxpPoCxpTurnOff
Category	CoaXPress
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Visibility	Expert
Values	-

Disable Power over CoaXPress (PoCXP) for the Link.

This feature shall be present only on receiver or transceiver Devices controlling PoCXP.
.

12.5.11 CxpPoCxpTripReset

Name	CxpPoCxpTripReset
Category	CoaXPress
Level	Optional
Interface	ICommand
Access	Write
Unit	-
Visibility	Expert
Values	-

Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

This feature shall be present only on receiver or transceiver Devices controlling PoCXP.

12.5.12 CxpPoCxpStatus

Name	CxpPoCxpStatus
Category	CoaXPress
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Expert
Values	Auto Off Tripped

Returns the Power over CoaXPress (PoCXP) status of the Device.

Possible values are:

- **Auto**: Normal automatic PoCXP operation.
- **Off**: PoCXP is forced off.
- **Tripped**: The Link has shut down because of an over-current trip.

This feature shall be present only on Receiver Device that controls the PoCXP.

13 User Set Control

This chapter 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.

13.1 UserSetControl

Name	UserSetControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category that contains the User Set control features.

13.2 UserSetSelector

Name	UserSetSelector
Category	UserSetControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Default UserSet0 (if 0 based), UserSet1, …

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

Possible values are:

- **Default**: Selects the factory setting user set.

- **UserSet0**: Selects the user set 0.
- **UserSet1**: Selects the user set 1.
- …

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

13.3 UserSetLoad

Name	UserSetLoad[UserSetSelector]
Category	UserSetControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Beginner
Values	-

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

13.4 UserSetSave

Name	UserSetSave[UserSetSelector]
Category	UserSetControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Beginner
Values	-

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

13.5 UserSetDefault

Name	UserSetDefault

Category	UserSetControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Default UserSet0 (if 0 based), UserSet1, …

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

Possible values are:

- **Default**: Select the factory setting user set.
- **UserSet0**: Select the user set 0.
- **UserSet1**: Select the user set 1.
- …

If **Default** is selected, the device will boot with the default factory settings and makes sure the continuous acquisition use case is ready to be used.

13.6 UserSetDefaultSelector (Deprecated)

Name	UserSetDefaultSelector
Category	UserSetControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	Default UserSet0 (if 0 based), UserSet1, …

This feature is deprecated (See UserSetDefault). Selects the feature User Set to load and make active when the device is reset.

To help backward compatibility, this feature can be included as Invisible in the device`s XML.

Possible values are:

- **Default**: Select the factory setting user set.
- **UserSet0**: Select the user set 0.
- **UserSet1**: Select the user set 1.
- …

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

13.7 UserSetFeatureSelector

Name	UserSetFeatureSelector
Category	UserSetControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Device-Specific

Selects which individual UserSet feature to control.

The feature lists all the features that can be a part of a device UserSet. All the device's UserSets have the same features.

Note that the name used in the enumeration must match exactly the device's feature name.

13.8 UserSetFeatureEnable

Name	UserSetFeatureEnable[UserSet FeatureSelector]
Category	UserSetControl
Level	Recommended
Interface	IBoolean
Access	Read/(Write)
Unit	-
Visibility	Expert

Values	True False

Enables the selected feature and make it active in all the UserSets.

14 Chunk Data Control

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.

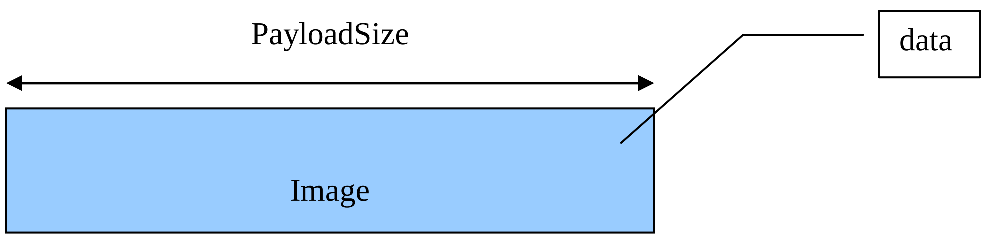


Figure 14-1: Frame with chunks disabled

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

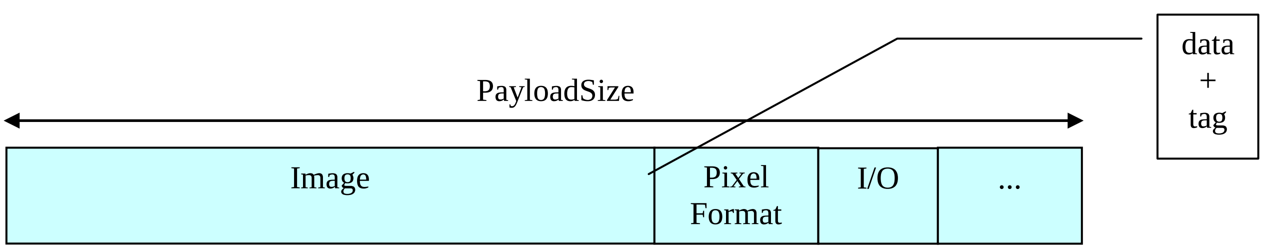


Figure 14-2: Frame with chunks enabled

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

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*.

14.1 ChunkDataControl

Name	ChunkDataControl

Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains the Chunk Data control features.

14.2 ChunkModeActive

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

Activates the inclusion of Chunk data in the payload of the image.

14.3 ChunkSelector

Name	ChunkSelector
Category	ChunkDataControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Image OffsetX

	OffsetY Width Height PixelFormat PixelDynamicRangeMax PixelDynamicRangeMin Timestamp LineStatusAll CounterValue TimerValue ExposureTime Gain BlackLevel LinePitchFrameID SourceID RegionID TransferBlockID TransferStreamID TransferQueueCurrentBlockCount StreamChannelID

Selects which Chunk to enable or control.

14.4 ChunkEnable

Name	ChunkEnable[ChunkSelector]
Category	ChunkDataControl
Level	Recommended
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Expert
Values	True False

Enables the inclusion of the selected Chunk data in the payload of the image.

14.5 ChunkImage

Name	ChunkImage

Category	ChunkDataControl
Level	Recommended
Interface	IRegister
Access	Read
Unit	-
Visibility	Guru
Values	Device-specific

Returns the entire image data included in the payload.

14.6 ChunkOffsetX

Name	ChunkOffsetX
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the **OffsetX** of the image included in the payload.

14.7 ChunkOffsetY

Name	ChunkOffsetY
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the **OffsetY** of the image included in the payload.

14.8 ChunkWidth

Name	ChunkWidth
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	>0

Returns the **Width** of the image included in the payload.

14.9 ChunkHeight

Name	ChunkHeight
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	>0

Returns the **Height** of the image included in the payload.

14.10 ChunkPixelFormat

Name	ChunkPixelFormat
Category	ChunkDataControl
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-

Visibility	Expert
Values	Mono1p Mono2p Mono4p Mono8 Mono8s Mono10 Mono10c3a64 Mono10c3p32 Mono10g12 Mono10msb Mono10p Mono10pmsb Mono10s Mono12 Mono12g Mono12msb Mono14 Mono16 R8 G8 B8 RGB8 RGB8_Planar RGB8a32 RGBa8 RGB10 RGB10_Planar RGB10g32 RGB10g32msb RGB10p32 RGB10p32msb RGB12 RGB12_Planar RGB16 RGB16_Planar RGB565p BGR10 BGR12 BGR16

	BGR565p BGR8 BGRa8 YUV411_8 YUV422_8 YUV8 YCbCr411_8 YCbCr422_8 YCbCr601_411_8 YCbCr601_422_8 YCbCr601_8 YCbCr709_411_8 YCbCr709_422_8 YCbCr709_8 YCbCr8 BayerBG8 BayerGB8 BayerGR8 BayerRG8 BayerBG10 BayerBG10g12 BayerGB10 BayerGB10g12 BayerGR10 BayerGR10g12 BayerRG10 BayerRG10g12 BayerBG12 BayerBG12g BayerGB12 BayerGB12g BayerGR12 BayerGR12g BayerRG12 BayerRG12g BayerBG16 BayerGB16 BayerGR16 BayerRG16 Raw16

	Raw8 Device-specific **- GigE Vision Specific:** Mono12Packed BayerGR10Packed BayerRG10Packed BayerGB10Packed BayerBG10Packed BayerGR12Packed BayerRG12Packed BayerGB12Packed BayerBG12Packed RGB10V1Packed BGR10V1Packed RGB12V1Packed **- Deprecated:** Mono8Signed (Deprecated, use Mono8s) RGB8Packed (Deprecated, use RGB8) BGR8Packed (Deprecated, use BGR8) RGBA8Packed (Deprecated, use RGBa8) BGRA8Packed (Deprecated, use BGRa8) RGB10Packed (Deprecated, use RGB10) BGR10Packed (Deprecated, use BGR10) RGB12Packed (Deprecated, use RGB12) BGR12Packed (Deprecated, use BGR12) RGB16Packed (Deprecated, use RGB16) BGR16Packed (Deprecated, use BGR16) RGB10V2Packed (Deprecated, use RGB10p32) BGR10V2Packed (Deprecated, use BGR10p32) RGB565Packed (Deprecated, use RGB565p) BGR565Packed (Deprecated, use BGR565p) YUV411Packed (Deprecated, use YUV411_8_UYYVYY) YUV422Packed (Deprecated, use YUV422_8_UYVY) YUV444Packed (Deprecated, use YUV8_UYV)

	YUYVPacked (Deprecated, use YUV422_8) RGB8Planar (Deprecated, use RGB8_Planar) RGB10Planar (Deprecated, use RGB10_Planar) RGB12Planar (Deprecated, use RGB12_Planar) RGB16Planar (Deprecated) use RGB16_Planar)

Returns the **PixelFormat** of the image included in the payload.

Possible values are:

- **Mono1p**: Mono 1 bit packed.
- **Mono2p**: Mono 2 bit packed.
- **Mono4p**: Mono 4 bit packed.
- **Mono8**: Mono 8 bit packed.
- **Mono8s**: Mono 1 bit signed.
- **Mono10**: Mono 10 bit.
- **Mono10c3a64**: Mono 10 bit in 64 bit.
- **Mono10c3p32**: Mono 10 bit in 32bit.
- **Mono10g12**: Mono 10 bit grouped in 12 bit.
- **Mono10msb**: Mono 10 bit packed aligned on Msb.
- **Mono10p**: Mono 10 bit packed.
- **Mono10pmsb**: Mono 10 bit packed aligned on Msb.
- **Mono10s**: Mono 10 bit signed.
- **Mono12**: Mono 12 bit packed.
- **Mono12g**: Mono 12 bit grouped.
- **Mono12msb**: Mono 12 bit aligned on Msb.
- **Mono14**: Mono 14 bit.
- **Mono16**: Mono 16 bit.
- **R8**: Red 8 bit.
- **G8**: Green 8 bit.
- **B8**: Blue 8 bit.

- **RGB8**: Red, Green, Blue 8 bit
- **RGB8_Planar**: Red, Green, Blue 8 bit planar.
- **RGB8a32:** Red, Green, Blue 8 bit aligned in 32 bit pixel
- **RGBa8:** Red, Green, Blue 8 bit aligned on 8 bit
- **RGB10:** Red, Green, Blue 10 bit.
- **RGB10_Planar:** Red, Green, Blue 10 bit planar.
- **RGB10g32:** Red, Green, Blue 8 bit grouped in 32 bit pixel.
- **RGB10g32msb:** Red, Green, Blue 10 bit grouped in 32 bit pixel aligned on Msb.
- **RGB10p32:** Red, Green, Blue 10 bit packed in 32 bit pixel.
- **RGB10p32msb:** Red, Green, Blue 10 bit packed in 32 bit pixel.
- **RGB12:** Red, Green, Blue 12 bit.
- **RGB12_Planar:** Red, Green, Blue 12 bit planar.
- **RGB16:** Red, Green, Blue 16 bit.
- **RGB16_Planar:** Red, Green, Blue 16 bit planar.
- **RGB565p:** Red, Green, Blue 16 bit packet in 5, 6, 5 bits.
- **BGR10:** Blue, Green, Red 10 bit.
- **BGR12:** Blue, Green, Red 12 bit.
- **BGR16:** Blue, Green, Red 16 bit.
- **BGR565p:** Blue, Green, Red 16 bit packet in 5, 6, 5 bits.
- **BGR8:** Blue, Green, Red 8 bit.
- **BGRa8:** Blue, Green, Red, Alpha 8 bit.
- **YUV411_8:** YUV 411, 8 bit.
- **YUV422_8:** YUV 422 8 bit.
- **YUV8:** YUV 8 bit.
- **YCbCr411_8:** YCrCb 411 8 bit.
- **YCbCr422_8:** YCrCb 422 8 bit.
- **YCbCr601_411_8:** YCrCb 601 411 8 bit.
- **YCbCr601_422_8:** YCrCb 601 422 8 bit.
- **YCbCr601_8:** YCrCb 601 8 bit.
- **YCbCr709_411_8:** YCrCb 709 411 8 bit.

- **YCbCr709_422_8:** YCrCb 709 422 8 bit.
- **YCbCr709_8:** YCrCb 709 8 bit.
- **YCbCr8:** YCbCr 8 bit.
- **BayerBG8:** Bayer Blue Green 8 bit.
- **BayerGB8:** Bayer Green Blue 8 bit.
- **BayerGR8:** Bayer Green Red 8 bit.
- **BayerRG8:** Bayer Red Green 8 bit.
- **BayerBG10:** Bayer Blue Green 10 bit.
- **BayerBG10g12:** Bayer Blue Green 8 bit grouped on 12 bit.
- **BayerGB10:** Bayer Green Blue 10 bit.
- **BayerGB10g12:** Bayer Green Blue 10 bit grouped on 12 bit.
- **BayerGR10:** Bayer Green Red 10 bit.
- **BayerGR10g12:** Bayer Green Red 10 bit grouped on 12 bit.
- **BayerRG10:** Bayer Red Green 10 bit.
- **BayerRG10g12:** Bayer Red Green 10 bit grouped on 12 bit.
- **BayerBG12:** Bayer Blue Green 12 bit
- **BayerBG12g:** Bayer Blue Green 12 bit grouped.
- **BayerGB12:** Bayer Green Blue 12 bit
- **BayerGB12g:** Bayer Green Blue 12 bit grouped on 12 bit.
- **BayerGR12:** Bayer Green Red 12 bit.
- **BayerGR12g:** Bayer Green Red 12 bit grouped on 12 bit.
- **BayerRG12:** Bayer Red Green 12 bit.
- **BayerRG12g:** Bayer Red Green 12 bit grouped on 12 bit.
- **BayerBG16:** Bayer Blue Green 16 bit.
- **BayerGB16:** Bayer Green Blue 16 bit.
- **BayerGR16:** Bayer Green Red 16 bit.
- **BayerRG16:** Bayer Red Green 16 bit.
- **Raw16:** Raw 16 bit.
- **Raw8:** Raw bit.
- **Mono12Packed:** Mono 12 bit packed (GigE Vision Specific).

- **BayerGR10Packed:** Bayer GR 10 bit packed (GigE Vision Specific).
- **BayerRG10Packed:** Bayer RG 10 bit packed (GigE Vision Specific).
- **BayerGB10Packed:** Bayer GB 10 bit packed (GigE Vision Specific).
- **BayerBG10Packed:** Bayer BG 10 bit packed (GigE Vision Specific).
- **BayerGR12Packed:** Bayer GR 12 bit packed (GigE Vision Specific).
- **BayerRG12Packed:** Bayer RG 12 bit packed (GigE Vision Specific).
- **BayerGB12Packed:** Bayer GB 12 bit packed (GigE Vision Specific).
- **BayerBG12Packed:** Bayer BG 12 bit packed (GigE Vision Specific).
- **RGB10V1Packed:** RGB 10 bit packed (GigE Vision Specific).
- **BGR10V1Packed:** BGR 10 bit packed (GigE Vision Specific).
- **RGB12V1Packed:** RGB 12 bit packed (GigE Vision Specific).

Note that only a subset of the possible pixel formats is listed here.
See the **PixelFormat** feature for more details.

14.11 ChunkPixelDynamicRangeMin

Name	ChunkPixelDynamicRangeMin
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the minimum value of dynamic range of the image included in the payload.

14.12 ChunkPixelDynamicRangeMax

Name	ChunkPixelDynamicRangeMax
Category	ChunkDataControl
Level	Recommended
Interface	IInteger

Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the maximum value of dynamic range of the image included in the payload.

14.13 ChunkTimestamp

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

Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event.

See Figure 5-3 for more details on FrameStart.

14.14 ChunkTimestampLatchValue

Name	ChunkTimestampLatchValue
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	ns
Visibility	Expert
Values	≥0

Returns the last Timestamp latched with the TimestampLatch command.

14.15 ChunkLineStatusAll

Name	ChunkLineStatusAll
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the status of all the I/O lines at the time of the FrameStart internal event.

14.16 ChunkCounterSelector

Name	ChunkCounterSelector
Category	ChunkDataControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Counter0 (If 0 based), Counter1, Counter2,…

Selects which counter to retrieve data from.

Possible values are:

- **Counter0**: Selects the counter 0.
- **Counter1**: Selects the counter 1.
- **Counter2**: Selects the counter 2.
- …

14.17 ChunkCounterValue

Name	ChunkCounterValue[ChunkCounterSelector]
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the value of the selected Chunk counter at the time of the FrameStart event.

14.18 ChunkTimerSelector

Name	ChunkTimerSelector
Category	ChunkDataControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Timer0 (If 0 based), Timer1, Timer2, …

Selects which Timer to retrieve data from.

Possible values are:

- **Timer0:** Selects the first Timer.
- **Timer1:** Selects the first Timer.
- **Timer2:** Selects the second Timer.
- **…**

14.19 ChunkTimerValue

Name	ChunkTimerValue[ChunkTimerSelector]
Category	ChunkDataControl
Level	Recommended
Interface	IFloat
Access	Read
Unit	us
Visibility	Expert
Values	>0

Returns the value of the selected Timer at the time of the FrameStart internal event.

14.20 ChunkExposureTime

Name	ChunkExposureTime
Category	ChunkDataControl
Level	Recommended
Interface	IFloat
Access	Read
Unit	us
Visibility	Expert
Values	≥0

Returns the exposure time used to capture the image.

14.21 ChunkGainSelector

Name	ChunkGainSelector
Category	ChunkDataControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-

Visibility	Expert
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, …

Selects which Gain to return.

Possible values 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.
- …

14.22 ChunkGain

Name	ChunkGain[ChunkGainSelector]
Category	ChunkDataControl
Level	Recommended
Interface	IFloat
Access	Read
Unit	-
Visibility	Expert
Values	Device-specific

Returns the gain used to capture the image.

14.23 ChunkBlackLevelSelector

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

Selects which Black Level to return. Possible values 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.
- …

14.24 ChunkBlackLevel

Name	ChunkBlackLevel[ChunkBlackLevelSelector]
Category	ChunkDataControl

Level	Recommended
Interface	IFloat
Access	Read
Unit	-
Visibility	Expert
Values	Device-specific

Returns the black level used to capture the image included in the payload.

14.25 ChunkLinePitch

Name	ChunkLinePitch
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	B
Visibility	Expert
Values	>0

Returns the **LinePitch** of the image included in the payload.

14.26 ChunkFrameID

Name	ChunkFrameID
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the unique Identifier of the frame (or image) included in the payload.

Recommended behavior of the identifier: It should start at a certain minimum value and keep incrementing by one for each frame up to a maximum, then it wraps to the minimum again. Each streaming channel should maintain the Frame ID separately.

Note: For GigE Vision, this chunk is not necessarily the block_id field included in the GVSP headers but can be equal to it.

14.27 ChunkSourceID

Name	ChunkSourceID
Category	ChunkDataControl
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-
Visibility	Expert
Values	Source0 (if 0 based) Source1 Source2 …

Returns the identifier of Source that the image comes from.

Possible values are:

- **Source0**: Image comes from the Source 0.
- **Source1**: Image comes from the Source 1.
- **Source2**: Image comes from the Source 2.
- …

14.28 ChunkRegionID

Name	ChunkRegionID
Category	ChunkDataControl
Level	Optional
Interface	IEnumeration
Access	Read
Unit	-

Visibility	Expert
Values	Region0 (if 0 based) Region1 Region2 …

Returns the identifier of Region that the image comes from.

Possible values are:

- **Region0**: Image comes from the Region 0.
- **Region1**: Image comes from the Region 1.
- **Region2**: Image comes from the Region 2.
- …

14.29 ChunkTransferBlockID

Name	ChunkTransferBlockID
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the unique identifier of the transfer block used to transport the payload.

The block ID is usually defined by the transport layer and repeated in the chunk for convenience.

14.30 ChunkTransferStreamID

Name	ChunkTransferStreamID
Category	ChunkDataControl
Level	Recommended
Interface	IEnumeration
Access	Read
Unit	-

Visibility	Expert
Values	Stream0 (if 0 based) Stream1 Stream2 Stream3 …

Returns identifier of the stream that generated this block.

Possible values are:

- **Stream0**: Data comes from Stream0.
- **Stream1**: Data comes from Stream1.
- **Stream2**: Data comes from Stream2.
- **Stream3**: Data comes from Stream3.

Note that the Stream used can be changed using the RegionDestination feature.

14.31 ChunkTransferQueueCurrentBlockCount

Name	ChunkTransferQueueCurrentBlockCount
Category	ChunkDataControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Returns the current number of blocks in the transfer queue.

14.32 ChunkStreamChannelID

Name	ChunkStreamChannelID
Category	ChunkDataControl
Level	Recommended
Interface	IInteger

Access	Read
Unit	-
Visibility	Expert
Values	>0

Returns identifier of the stream channel used to carry the block.

Note that the Stream Channel used can be changed using the TransferStreamChannel feature.

14.33 ChunkSequencerSetActive

Name	ChunkSequencerSetActive
Category	ChunkDataControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Return the index of the active set of the running sequencer included in the payload.

15 File Access Control

The File Access Controls chapter 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:

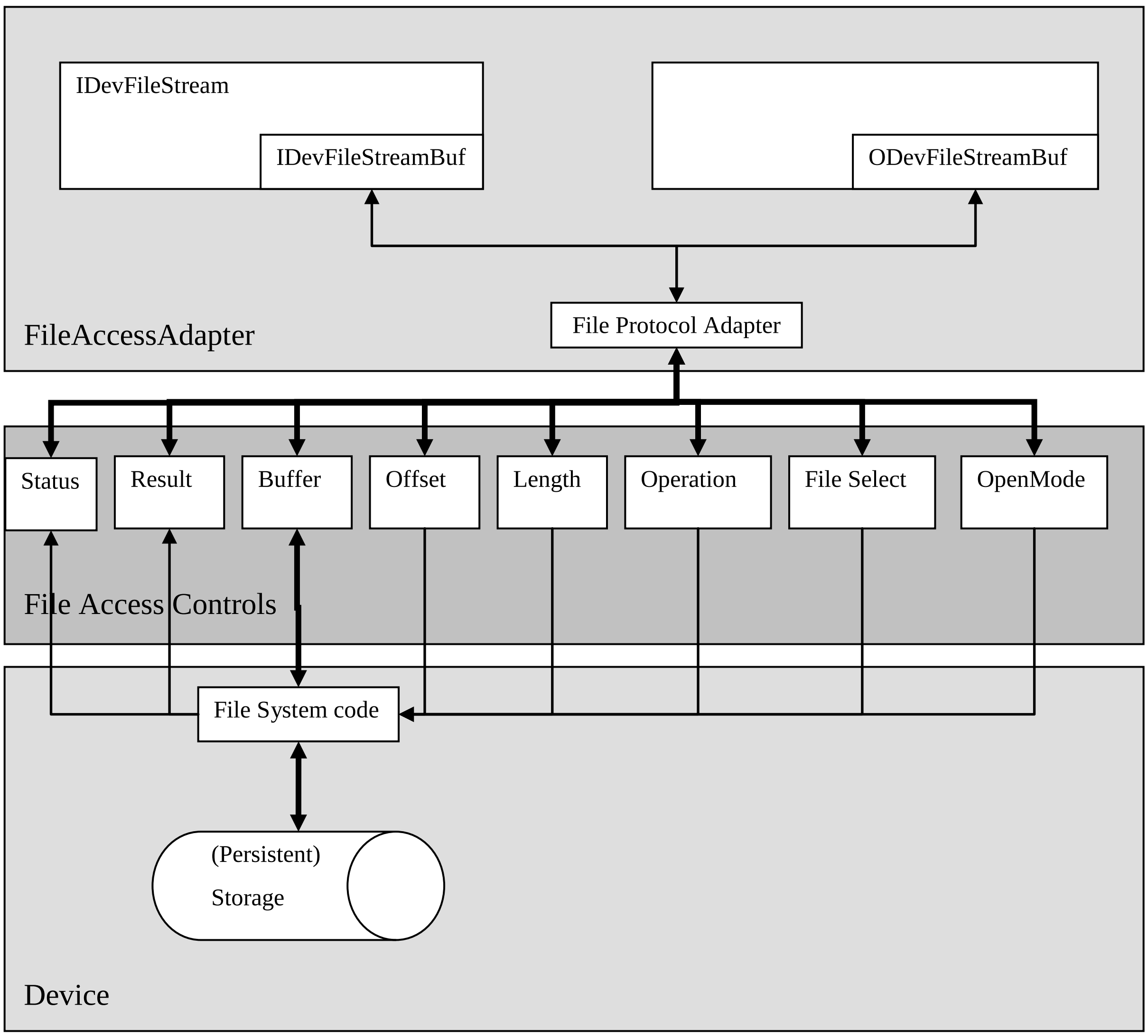


Figure 15-1: File Access Model

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:

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 executing 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**.

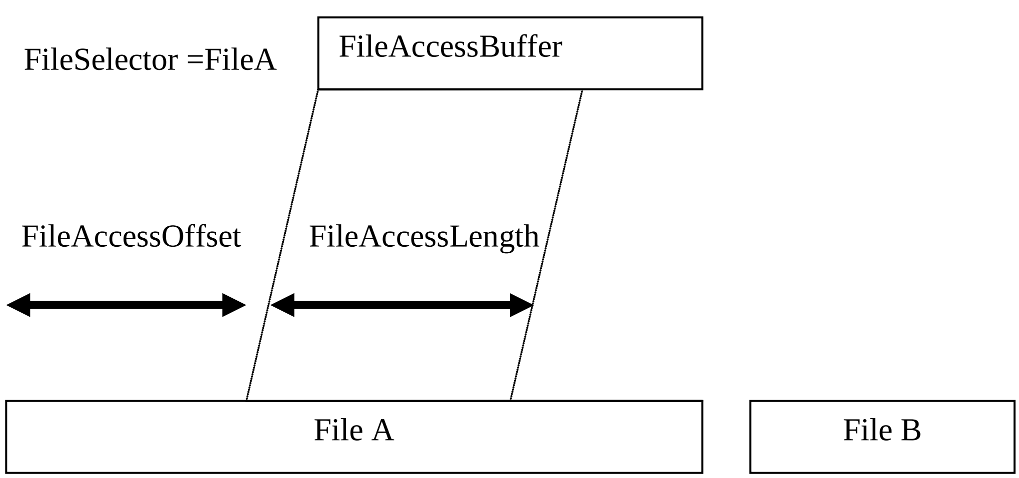


Figure 15-2: Layout of File Access Buffer.

15.1 FileAccessControl

Name	FileAccessControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Guru
Values	-

Category that contains the File Access control features.

15.2 FileSelector

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

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.

Possible values are:

- **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.

15.3 FileOperationSelector

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

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

Possible values are:

- **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.**
- **Delete:** Deletes the file selected by **FileSelector** in the device. Note that deleting a device file should not remove the associated FileSelector entry to allow future operation on this file.

15.4 FileOperationExecute

Name	FileOperationExecute[FileSelector][FileOperationSelector]
Category	FileAccessControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Guru
Values	-

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

15.5 FileOpenMode

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

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

Possible values are:

- **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.

15.6 FileAccessBuffer

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

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 executing 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).

15.7 FileAccessOffset

Name	FileAccessOffset[FileSelector][FileOperationSelector]
Category	FileAccessControl
Level	Recommended
Interface	IInteger
Access	Read/(Write)
Unit	B
Visibility	Guru

Values	≥ 0

Controls the Offset of 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.

15.8 FileAccessLength

Name	FileAccessLength[FileSelector][FileOperationSelector]
Category	FileAccessControl
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	B
Visibility	Guru
Values	≥ 0

Controls the Length of 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 Figure 14-2). This feature is available only when **FileOperationSelector** is set to Read or Write.

15.9 FileOperationStatus

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

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**.

Possible values are:

- **Success:** File Operation was sucessful.
- **Failure:** File Operation failed.

15.10 FileOperationResult

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

Represents the file operation result. For Read or Write operations, the number of successfully read/written bytes is returned.

15.11 FileSize

Name	FileSize[FileSelector]
Category	FileAccessControl
Level	Recommended
Interface	IInteger
Access	Read
Unit	B
Visibility	Guru
Values	≥0

Represents the size of the selected file in bytes.

16 Color Transformation Control

The Color Transformation chapter describes all features related to color Transformations in the device.

The Color Transformation is a linear operation taking as input a triplet of Components (C0, C1, C2) for a color pixel (Typicaly: R_{in}, G_{in}, B_{in} representing a RGB color pixel). This triplet is first multiplied by a 3x3 matrix and then added to an offset triplet.

The equation is given by:

$$\begin{pmatrix} C0_{out} \\ C1_{out} \\ C2_{out} \end{pmatrix} = \begin{pmatrix} Gain00 & Gain01 & Gain02 \\ Gain10 & Gain11 & Gain12 \\ Gain20 & Gain21 & Gain22 \end{pmatrix} \begin{pmatrix} C0_{in} \\ C1_{in} \\ C2_{in} \end{pmatrix} + \begin{pmatrix} Offset0 \\ Offset1 \\ Offset2 \end{pmatrix}$$

Equivalent: $$\begin{pmatrix} R_{out} \\ G_{out} \\ B_{out} \end{pmatrix} = \begin{pmatrix} RR & RG & RB \\ GR & GG & GB \\ BR & BG & BB \end{pmatrix} \begin{pmatrix} R_{in} \\ G_{in} \\ B_{in} \end{pmatrix} + \begin{pmatrix} R_{offset} \\ G_{offset} \\ B_{offset} \end{pmatrix}$$

The descriptions below assume RGB to RGB transformation:

Where	$C0_{in}$ is the first component of the incoming pixel
	$C1_{in}$ is the second component of the incoming pixel
	$C2_{in}$ is the third component of the incoming pixel
	Gain00 is the red contribution to the red pixel (multiplicative factor)
	Gain01 is the green contribution to the red pixel (multiplicative factor)
	Gain02 is the blue contribution to the red pixel (multiplicative factor)
	Gain10 is the red contribution to the green pixel (multiplicative factor)
	Gain11 is the green contribution to the green pixel (multiplicative factor)
	Gain12 is the blue contribution to the blue pixel (multiplicative factor)
	Gain20 is the red contribution to the blue pixel (multiplicative factor)
	Gain21 is the green contribution to the blue pixel (multiplicative factor)
	Gain22 is the blue contribution to the blue pixel (multiplicative factor)
	Offset0 is the red offset

	Offset1 is the green offset
	Offset2 is the blue offset
	$C0_{out}$ is the first resulting component of the pixel after the transformation
	$C1_{out}$ is the second resulting component of the pixel after the transformation
	$C2_{out}$ is the third resulting component of the pixel after the transformation

Example for YUV convertion:

The Color Transformation can also be used outside of the simple scope of color correction on RGB pixels. For instance, it can be used as a color convert to convert RGB to YUV.

Here is the example of this conversion for 8-bit pixels:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R_{in} \\ G_{in} \\ B_{in} \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

16.1 ColorTransformationControl

Name	ColorTransformationControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains the Color Transformation control features.

16.2 ColorTransformationSelector

Name	ColorTransformationSelector
Category	ColorTransformationControl
Level	Optional
Interface	IEnumeration

Access	Read/Write
Unit	-
Visibility	Expert
Values	RGBtoRGB RGBtoYUV Device-specific

Selects which Color Transformation module is controlled by the various Color Transformation features.

Possible values are:

- **RGBtoRGB:** RGB to RGB color transformation.
- **RGBtoYUV:** RGB to YUV color transformation.

It is typically not available when a single Color Transformation module is supported.

16.3 ColorTransformationEnable

Name	ColorTransformationEnable[ColorTransformationSelector]
Category	ColorTransformationControl
Level	Optional
Interface	IBoolean
Access	Read/Write
Unit	-
Visibility	Expert
Values	True False

Activates the selected Color Transformation module.

16.4 ColorTransformationValueSelector

Name	ColorTransformationValueSelector[ColorTransformationSelector]
Category	ColorTransformationControl
Level	Optional
Interface	IEnumeration

Access	Read/Write
Unit	-
Visibility	Expert
Values	Gain00 Gain01 Gain02 Gain10 Gain11 Gain12 Gain20 Gain21 Gain22 Offset0 Offset1 Offset2

Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module.

Possible values are:

- **Gain00:** Gain 0,0 of the transformation matrix.
- **Gain01:** Gain 0,1 of the transformation matrix.
- **Gain02:** Gain 0,2 of the transformation matrix.
- **Gain10:** Gain 1,0 of the transformation matrix.
- **Gain11:** Gain 1,1 of the transformation matrix.
- **Gain12:** Gain 1,2 of the transformation matrix.
- **Gain20:** Gain 2,0 of the transformation matrix.
- **Gain21:** Gain 2,1 of the transformation matrix.
- **Gain22:** Gain 2,2 of the transformation matrix.
- **Offset0:** Offset 0 of the transformation matrix.
- **Offset1:** Offset 1 of the transformation matrix.
- **Offset2:** Offset 2 of the transformation matrix.

16.5 ColorTransformationValue

Name	ColorTransformationValue[ColorTransformationSelector][ColorTransformationValueSelector]
Category	ColorTransformationControl
Level	Optional
Interface	IFloat
Access	Read/Write
Unit	-
Visibility	Expert
Values	Device-Specific

Represents the value of the selected Gain factor or Offset inside the Transformation matrix.

17 Action Control

The Action chapter describes all features related to Action Signals in the device.

Action Signals are a method to trigger actions in multiple devices at the same time (depending on the specific transport layer). Action Signals are used in the device in the same way as e.g. digital input lines.

One possible use for action signals is to raise a FrameStart trigger in multiple devices at the same time.

On most transport layers Action Signals are implemented using broadcast protocol messages. To allow a finegrained control which devices are allowed to react on the broadcasted action protocol message, the features **ActionDeviceKey**, **ActionGroupKey** and **ActionGroupMask** define filter conditions.

Each action protocol message contains an action device key, action group key and an action group mask. If the device detects a match between this protocol information and one of the actions selected by **ActionSelector** it raises the corresponding Action Signal.

See chapter 22.4 "Action Signal Examples" for typical uses cases.

17.1 ActionControl

Name	ActionControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Guru
Values	-

Category that contains the Action control features.

17.2 ActionUnconditionalMode

Name	ActionUnconditionalMode
Category	ActionControl
Level	Optional

Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	Off On

Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

Possible values are:

- **Off:** Unconditional mode is disabled.
- **On:** Unconditional mode is enabled.

17.3 ActionDeviceKey

Name	ActionDeviceKey
Category	ActionControl
Level	Optional
Interface	IInteger
Access	Write-Only
Unit	-
Visibility	Guru
Values	≥ 0

Provides the device key that allows the device to check the validity of action commands. The device internal assertion of an action signal is only authorized if the **ActionDeviceKey** and the action device key value in the protocol message are equal.

17.4 ActionQueueSize

Name	ActionQueueSize
Category	ActionControl
Level	Optional
Interface	IInteger
Access	Read

Unit	-
Visibility	Guru
Values	≥ 0

Indicates the size of the scheduled action commands queue. This number represents the maximum number of scheduled action commands that can be pending at a given point in time.

17.5 ActionSelector

Name	ActionSelector
Category	ActionControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru
Values	>0

Selects to which Action Signal further Action settings apply.

17.6 ActionGroupMask

Name	ActionGroupMask[ActionSelector]
Category	ActionControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru
Values	≥0

Provides the mask that the device will use to validate the action on reception of the action protocol message.

The device asserts the selected Action signal only if:

- The selected **ActionDeviceKey** is equal to the action device key in the action protocol message.
- The logical AND-wise operation of the action group mask in the action protocol message against the selected **ActionGroupMask** is non-zero.
- The selected **ActionGroupKey** is equal to the action group key in the action protocol message.

17.7 ActionGroupKey

Name	ActionGroupKey[ActionSelector]
Category	ActionControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru
Values	≥0

Provides the key that the device will use to validate the action on reception of the action protocol message.

The device asserts the selected Action signal only if:

- The selected **ActionDeviceKey** is equal to the action device key in the action protocol message.
- The logical AND-wise operation of the action group mask in the action protocol message against the selected **ActionGroupMask** is non-zero.
- The selected **ActionGroupKey** is equal to the action group key in the action protocol message.

18 Source Control

The Source Control chapter describes the features related to multi-source acquisition devices. An example of such a multi-source device would be a camera with 2 independent sensors (visible and infra red light) that would transmit their data on a single camera`s physical connector but would have independent controllable features.

18.1 Sources Control model with Multi-Regions and Transfer

This section explains the general model for control of the acquisition on multi-source devices.

The section also presents the relation of the sources with other elements such as the region of interest and the data transfer control module that can be found in more sophisticated multi-source devices.

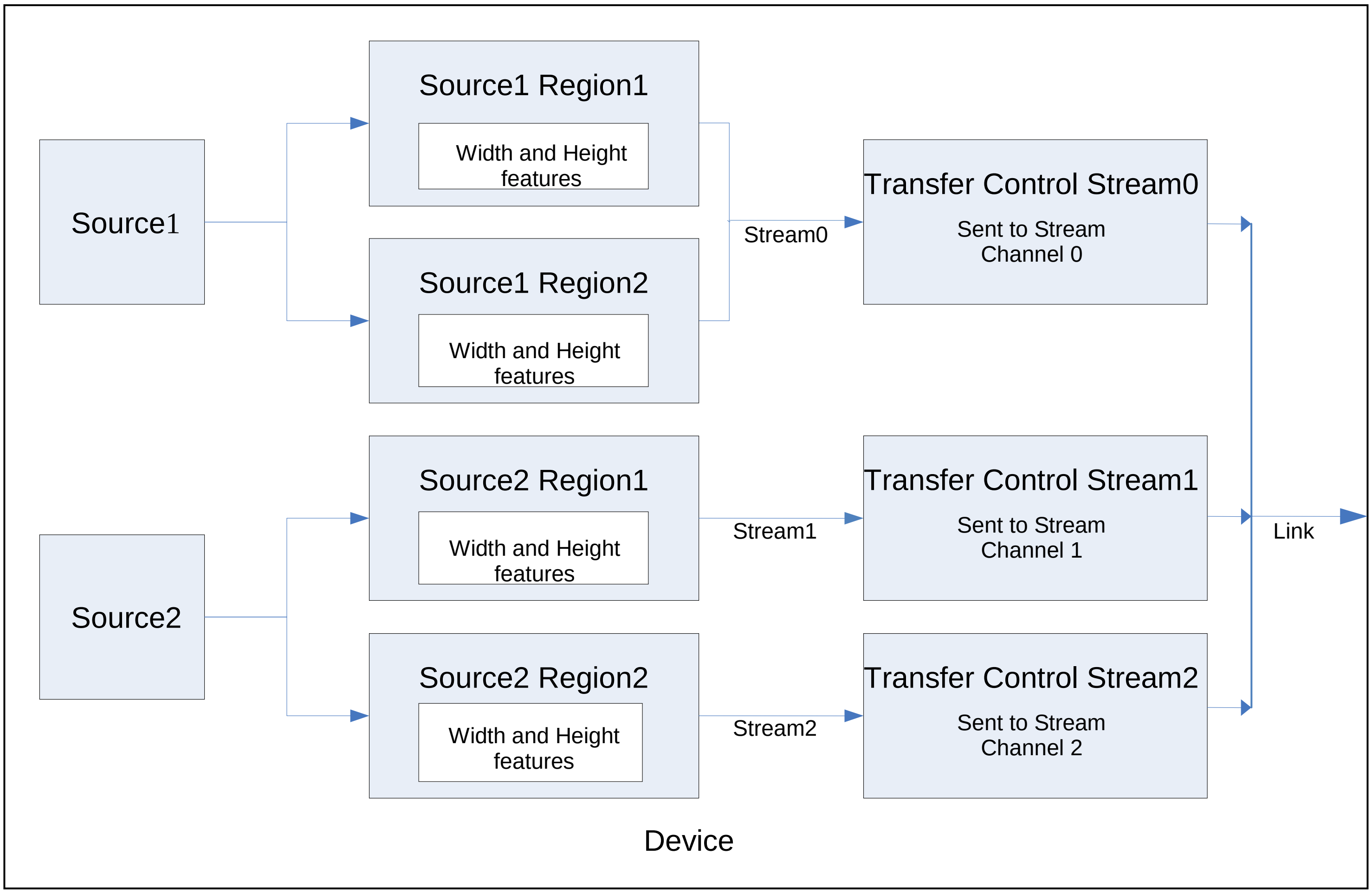

Source1 Region1
Width and Height features
Source1
Source1 Region2
Width and Height features
Stream0
Transfer Control Stream0
Sent to Stream Channel 0
Source2 Region1
Width and Height features
Stream1
Transfer Control Stream1
Sent to Stream Channel 1
Link
Source2
Source2 Region2
Width and Height features
Stream2
Transfer Control Stream2
Sent to Stream Channel 2
Device

Figure 18-1: Multi-Source multi-Region Device with data stream Transfer control.

Multi-Source Features setting example

Figure 18-1 above presents an example of a relatively complex device supporting multi-sources, multi-regions with data stream transfer control. This device features two regions of interest for each video source. In this particular example, both regions of interest from Source1 are streamed on the same output stream channel, while a separate stream channel is used for each region of interest of Source2. The three stream channels are transmitted out of the device using the same external Link to the target Host System.

The features setting for this use case is presented below. The detailed description of the feature to control the Sources can be found in section 18.2: "Source Control features below. The features for Region of interest and image format and handling are documented in chapter 4: "Image Format Control" and chapter 19: "Transfer Control" presents the features for explicit control of data Transfer.

So for the particular use case illustrated in Figure 18-1: above, the features would be set to:

Source 1, Region 1:

SourceSelector = Source1
RegionSelector[SourceSelector] = Region1
RegionMode[SourceSelector][RegionSelector] = On
RegionDestination[SourceSelector][RegionSelector] = Stream0
Width[SourceSelector] [RegionSelector] = 320
Height[SourceSelector][RegionSelector] = 240

Source 1, Region 2:

SourceSelector = Source1
RegionSelector[SourceSelector] = Region2
RegionMode[SourceSelector][RegionSelector] = On
RegionDestination[SourceSelector][RegionSelector] = Stream0
Width[SourceSelector] [RegionSelector] = 420
Height[SourceSelector][RegionSelector] = 340

Source 1, Region 1 and 2, Transfer and Acquisition control:

TransferSelector = Stream0
TransferControlMode[TransferSelector] = UserControlled
TransferStreamChannel[TransferSelector] = 0
TransferStart[TransferSelector]
AcquisitionStart[SourceSelector]
…
AcquisitionStop[SourceSelector]

TransferStop[TransferSelector]

Source 2 Region 1:

SourceSelector = Source2
RegionSelector[SourceSelector] = Region1
RegionMode[SourceSelector][RegionSelector] = On
RegionDestination[SourceSelector][RegionSelector] = Stream1
Width[SourceSelector] [RegionSelector] = 220
Height[SourceSelector][RegionSelector] = 140
TransferSelector = Stream1
TransferControlMode[TransferSelector] = UserControlled
TransferStreamChannel[TransferSelector] = 0

Source 2 Region 2:

SourceSelector = Source2
RegionSelector[SourceSelector] = Region2
RegionMode[SourceSelector][RegionSelector] = On
RegionDestination[SourceSelector][RegionSelector] = Stream2
Width[SourceSelector] [RegionSelector] = 220
Height[SourceSelector][RegionSelector] = 330
TransferSelector = Stream2
TransferControlMode[TransferSelector] = UserControlled
TransferStreamChannel[TransferSelector] = 0

Source 1, Region 1 and 2, Transfer and Acquisition control:

TransferSelector = Stream1
TransferStart[TransferSelector]
TransferSelector = Stream2
TransferStart[TransferSelector]
SourceSelector = Source2
AcquisitionStart[SourceSelector]
…
AcquisitionStop[SourceSelector]
TransferSelector = Stream1
TransferStop[TransferSelector]
TransferSelector = Stream2
TransferStop[TransferSelector]

18.2 Source Control features

This section describes the features related to the devices that support multiple video sources that are transmitted over a single link. The virtual Stream channels of a Link are used to transport the different video sources over a common physical connection.

The main feature in this section is the source selector feature (SourceSelector). This feature enables the features associated to a given video source to be controlled on a per video source basis even if they pertain to different feature categories. For instance, it can enable a user to independently set the Width feature (Image Format Control category) and the Gain feature (Analog Control category) for the two sources supported by a given device.

An example of independent features setting for a dual source device would be:

SourceSelector = Source1
Width[SourceSelector] = 320
Gain[SourceSelector] = 60
AcquisitionStart[SourceSelector]
…
AcquisitionStop[SourceSelector]

SourceSelector = Source2
Width[SourceSelector] = 240
Gain[SourceSelector] = 90
AcquisitionStart[SourceSelector]
…
AcquisitionStop[SourceSelector]

Features selected by the Source Selector Feature

The source selector feature can be an optional selector to a number of features defined in this document based on the specificity of product it represents.

In order to simplify the standard text and feature descriptions, the optional source selector is not propagated to all the features of the SFNC that it can potentially select. Table 9 summarizes which features could potentially be selected by the source selector.

Table 9: Source Selector Potential Selectees

Categories	Potential Selectees
Device Control	The DeviceScanType feature can potentially be selected by the source selector feature. **Note:** The DeviceTemperatureSelector, DeviceClockSelector and DeviceSerialPortSelector features may have more enumeration

Categories	Potential Selectees
	entries.
Image Format Control Acquisition Control Analog Control LUT Control Color Transformation Control	All features of these categories could potentially be selected by the source selector based on the product specificities.
Digital I/O Control Counter and Timer Control	The features of these categories are likely not selected by the source selector but could be depending on the particular device implementation. **Note:** The LineSelector, LineSource, CounterEventSource, CounterResetSource, CounterTriggerSource and TimerTriggerSource features may have more enumeration entries.
Event Control	The EventSelector feature could have more enumeration entries. In this case, the name of the source would be prepended to an existing enumeration entry name. For instance, the Source1AcquisitionStart and Source2AcquisitionStart enumeration entries can be added to enable the generation of events upon acquisition start for Source1 and Source2 respectively.
GenICam Control	The TLParamsLocked feature could potentially be selected by the source selector feature.
Transport Layer Control	The ClConfiguration, ClTimeSlotsCount and DeviceTapGeometry feature could potentially be selected by the source selector feature.
User Set Control	The features of this category are likely not selected by the source selector feature.
Chunk Data Control	The ChunkSelector feature could have more enumeration entries.
File Access Control	The FileSelector feature could have more enumeration entries.
Action Control	No features can be selected by the source selector feature.

18.3 SourceControl

Name	SourceControl
Category	Root
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category that contains the source control features.

18.4 SourceCount

Name	SourceCount
Category	SourceControl
Level	Optional
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	≥ 1

Controls or returns the number of sources supported by the device.

This feature is generally read-only but can be writable if the number of sources supported by the device is run-time programmable.

18.5 SourceSelector

Name	SourceSelector
Category	SourceControl
Level	Optional
Interface	IEnumeration
Access	Read/Write

Unit	-
Visibility	Expert
Values	Source0 (If 0 based) Source1 Source2 … All Device-specific

Selects the source to control.

Possible values are:

- **Source0**: Selects the data source 0.
- **Source1**: Selects the data source 1.
- **Source2**: Selects the data source 2.
- …
- **All**: Selects all the data sources.

The "All" value can be used to change the features of all the sources at the same time. For example, this can be useful to simultaneously start and stop multiple acquisitions.

19 Transfer Control

The Transfer Control chapter describes the model and features related to the explicit control of the transfer of the data acquired by a device.

A general model of the acquisition of images by a device and the role of the Transfer Control module was already presented in the section 1.6 "Device Acquisition Model" and illustared in detail in the section 18.1 "Sources Control model with Multi-Regions and Transfer".

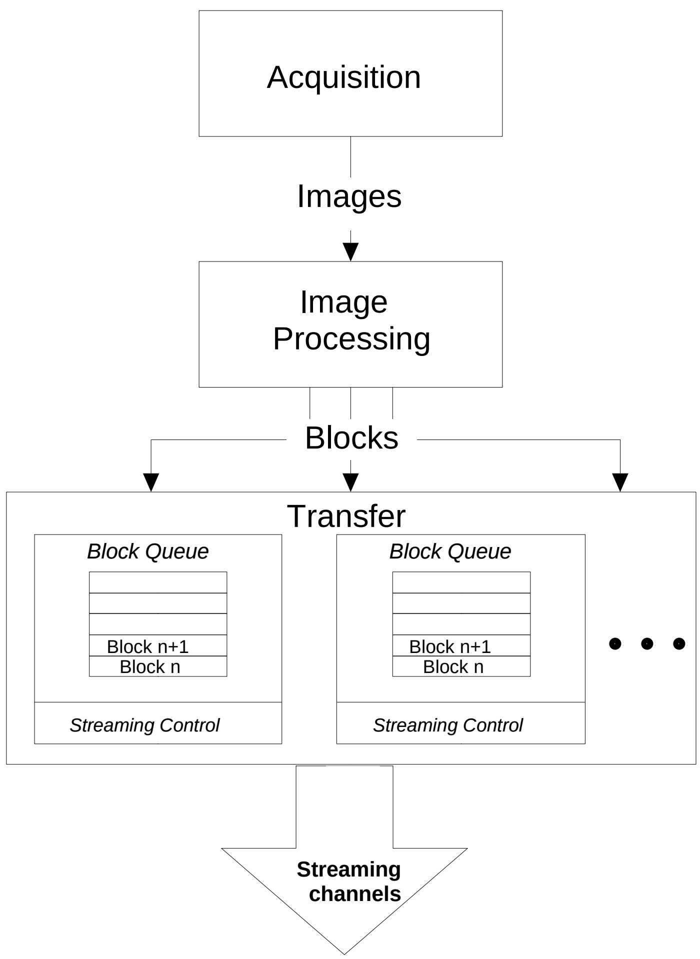
So this chapter will concentrates only on describing the Transfer mechanism itself.

19.1 Transfer Control Model

The Transfer Control Model section describes the features related to the transfer of data by the device. It describes the basic Transfer model and the typical behavior of the device when sending data to the outside.

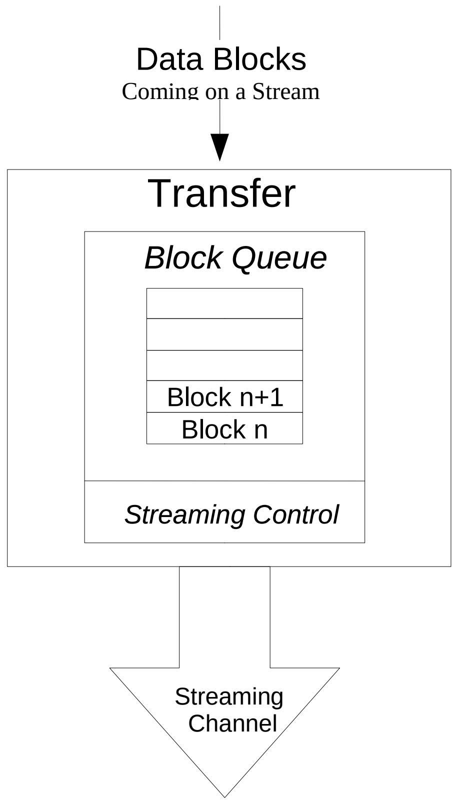
An acquisition typically generates images (or frames). Those images can be preprocessed (Ex: Bayer conversion) before transferrimg them out by the Device. In certain cases, those images can also be transformed in other type of data (such as an intensity histogram) by an internal image processing module. So it is possible that in addition to the original image acquired, a transformed image or some related data also need to be transferred out of the device. In the following model, it is considered that the captured images are transformed into different data blocks by an optional image processing module. Those data blocks are then sent to a transfer module on different data streams. The Transfer module will then transmit those data blocks externally on one or many streaming channels. This typical acquisition data flow is represented in Figure 19-1: Acquisition and Transfer data flow.

Figure 19-1: Acquisition and Transfer data flow.



In summary, a transfer is the action of streaming data blocks to another device. Data blocks are complex data structures that can represent images, image processing results or even files. The transfer module is composed of one or many block Queue(s) and Streaming Controls section(s) (see Figure 19-2.).

Figure 19-2: Transfer control section.



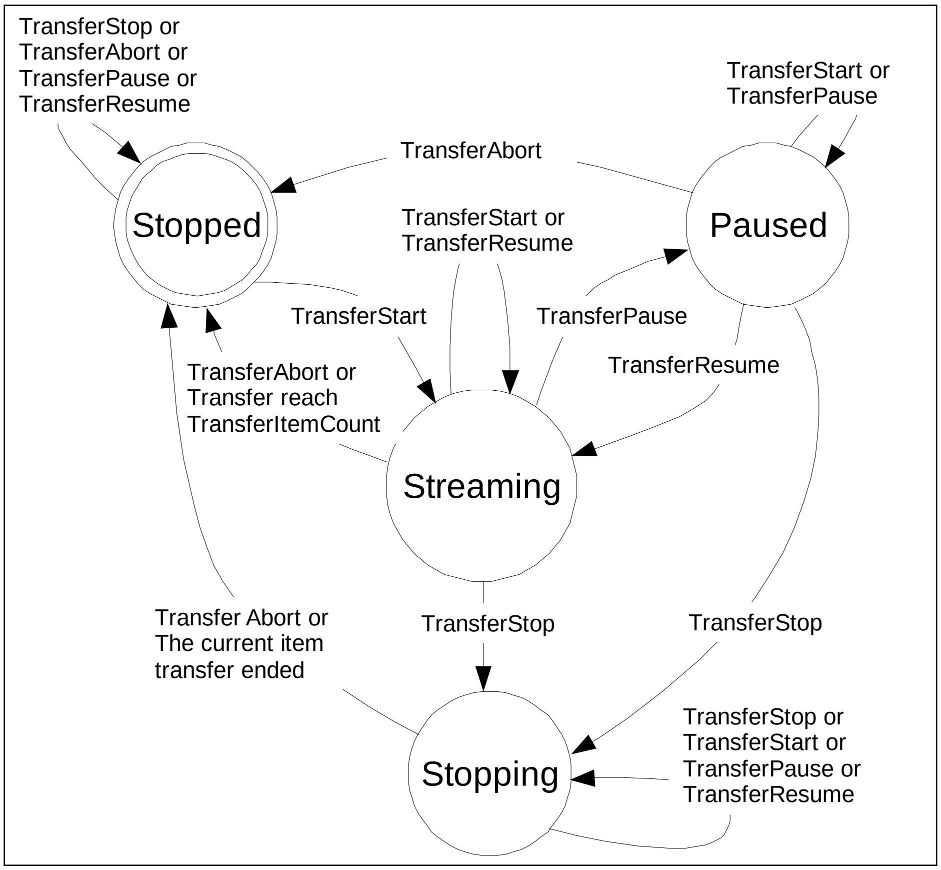
Block Queue

A Block Queue is used to store data blocks for the time interval between its generation and its transmission.

Streaming Control

The streaming control regulates the outgoing flow of data. The streaming can be in one of the following states: ***Stopped***, ***Stopping, Streaming*** and ***Paused***. The transfer module always accepts the new blocks of data from the image processing module regardless of the streaming state. The transfer control features **TransferStart**, **TransferStop**, **TransferAbort**, **TransferPause** and **TransferResume** allow the user to change the state of the streaming.

Figure 19-3: Transfer control state.



Streaming: Data blocks will be transmitted when enough data is available.

Stopping: Data blocks transmission is stopping. The current block transmission will be completed and the transfer state will change to ***Stopped.***

Stopped: Data blocks transmission is stopped.

Paused: Data blocks transmission is suspended immediately.

19.2 Transfer Control features

This section describes in detail the features related to the Transfer Control.

A detailed example of the usage of the Transfer Control features is presented in the section 18.1 "Sources Control model with Multi-Regions and Transfer".

19.3 TransferControl

Name	TransferControl
Category	Root
Level	Recommended
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category for the data Transfer Control features.

19.4 TransferSelector

Name	TransferSelector
Category	TransferControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Stream0 (if 0 based), Stream1, Stream2, … All

Selects which stream transfers are currently controlled by the selected Transfer features.

Possible values are:

- **Stream0**: The transfer features control the data stream 0.
- **Stream1**: The transfer features control the data stream 1.
- **Stream2**: The transfer features control the data stream 2.
- …
- **All**: The transfer features control all the data streams simulateneously.

19.5 TransferControlMode

Name	TransferControlMode[TransferSelector]
Category	TransferControl
Level	Recommended
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Basic Automatic UserControlled

Selects the control method for the transfers.

Possible values are:

- **Basic:** Transfer flow control mechanism is disabled. The **TransferStart**, **TransferPause**, **TransferResume**, **TransferStop** and **TransferAbort** features are not available to the user. This transfer mode is used to ensure compatibility with devices not aware of the transfer flow control mechanism.
- **Automatic**: Transfer flow control mechanism is controlled automatically. The Transfer features are controlled transparently by the acquisition features.
    - **TransferStart** is called during the AcquisitionStart.
    - **TransferStop** is never called.
    - **TransferAbort** is called during the AcquisitionAbort.
    - **TransferOperationMode** is read only and set to "Continuous".

  If available, the **TransferPause** and **TransferResume** features are controlled by the user.

- **UserControlled**: Transfer flow control mechanism is controlled by the user. The **TransferMode**, **TransferStart**, **TransferStop**, **TransferAbort**, **TransferPause** and **TransferResume** features are used to control manually the flow of data. In this mode, the features **TransferOperationMode**, **TransferStart** and **TransferStop** must be available.

If this feature is not present, the transfer control is assumed to be "Basic".

Note that the Transfers can also be controlled using external trigger signals (See TransferTriggerSelector).

19.6 TransferOperationMode

Name	TransferOperationMode[TransferSelector]
Category	TransferControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	Continuous MultiBlock

Selects the operation mode of the transfer.

Possible values are:

- **Continuous**: Blocks of data are transferred continuously until stopped with the **TransferStop** command.
- **MultiBlock**: The transfer of the blocks of data terminates automatically after the transmission of **TransferBlockCount** or when an explicit **TransferStop** command is received.

If this feature is not present, the transfer mode is assumed to be "Continuous".

19.7 TransferBlockCount

Name	TransferBlockCount[TransferSelector]
Category	TransferControl
Level	Optional
Interface	IInteger
Access	Read/(Write)

Unit	-
Visibility	Expert
Values	> 0

Specifies the number of data Blocks that the device should stream before stopping. This feature is only active if the **TransferOperationMode** is set to MultiBlock.

19.8 TransferBurstCount

Name	TransferBurstCount
Category	TransferControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥1

Number of Block(s) to transfer for each TransferBurstStart trigger.

This feature is used only if the TransferBurstStart trigger is enabled and the TransferBurstEnd trigger is disabled.

19.9 TransferQueueMaxBlockCount

Name	TransferQueueMaxBlockCount[TransferSelector]
Category	TransferControl
Level	Optional
Interface	IInteger
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	> 0

Controls the maximum number of data blocks that can be stored in the block queue of the selected stream.

19.10 TransferQueueCurrentBlockCount

Name	TransferQueueCurrentBlockCount[TransferSelector]
Category	TransferControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	-

Returns the number of Block(s) currently in the transfer queue.

19.11 TransferQueueMode

Name	TransferQueueMode[TransferSelector]
Category	TransferControl
Level	Optional
Interface	IEnumeration
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	FirstInFirstOut …

Specifies the operation mode of the transfer queue.

Possible values are:

- **FirstInFirstOut:** Blocks first In are transferred Out first.
- **…**

19.12 TransferStart

Name	TransferStart[TransferSelector]

Category	TransferControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Starts the streaming of data blocks out of the device. This feature must be available when the **TransferControlMode** is set to "UserControled". If the **TransferStart** feature is not writable (locked), the application should not start the transfer and should avoid using the feature until it becomes writable again.

19.13 TransferStop

Name	TransferStop[TransferSelector]
Category	TransferControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Stops the streaming of data Block(s). The current block transmission will be completed. This feature must be available when the **TransferControlMode** is set to "UserControlled".

19.14 TransferAbort

Name	TransferAbort[TransferSelector]
Category	TransferControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-

Visibility	Expert
Values	-

Aborts immediately the streaming of data block(s). Aborting the transfer will result in the lost of the data that is present or currently entering in the block queue. However, the next new block received will be stored in the queue and transferred to the host when the streaming is restarted. If implemented, this feature should be available when the **TransferControlMode** is set to "UserControlled".

19.15 TransferPause

Name	TransferPause[TransferSelector]
Category	TransferControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Guru
Values	-

Pauses the streaming of data Block(s). Pausing the streaming will immediately suspend the ongoing data transfer even if a block is partially transfered. The device will resume its transmission at the reception of a **TransferResume** command.

19.16 TransferResume

Name	TransferResume[TransferSelector]
Category	TransferControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Guru
Values	-

Resumes a data Blocks streaming that was previously paused by a **TransferPause** command.

19.17 TransferTriggerSelector

Name	TransferTriggerSelector[TransferSelector]
Category	TransferControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	TransferStart TransferStop TransferAbort TransferPause TransferResume TransferActive TransferBurstStart TransferBurstStop

Selects the type of transfer trigger to configure.

Possible values are:

- **TransferStart**: Selects a trigger to start the transfers.
- **TransferStop**: Selects a trigger to stop the transfers.
- **TransferAbort**: Selects a trigger to abort the transfers.
- **TransferPause**: Selects a trigger to pause the transfers.
- **TransferResume**: Selects a trigger to Resume the transfers.
- **TransferActive**: Selects a trigger to Activate the transfers. This trigger type is used when TriggerActivation is set LevelHigh or levelLow.
- **TransferBurstStart**: Selects a trigger to start the transfer of a burst of frames specified by TransferBurstCount.
- **TransferBurstStop**: Selects a trigger to end the transfer of a burst of frames.

19.18TransferTriggerMode

Name	TransferTriggerMode[TransferSelector][TransferTriggerSelector]
Category	TransferControl
Level	Recommended

Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	Off On

Controls if the selected trigger is active.

Possible values are:

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

19.19 TransferTriggerSource

Name	TransferTriggerSource[TransferTriggerSelector]
Category	TransferControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	Line0 (If 0 based), Line1, Line2, … Counter0Start (If 0 based), Counter1Start, Counter2Start, … Counter0End (If 0 based), Counter1End, Counter2End, … Timer0Start (If 0 based), Timer1Start, Timer2Start, … Timer0End (If 0 based), Timer1End, Timer2End, … SoftwareSignal0 (If 0 based), SoftwareSignal1, SoftwareSignal2, … Action0 (If 0 based), Action1, Action2, …

Specifies the signal to use as the trigger source for transfers.

Possible values are:

- **Line0** (If 0 based)**, Line1, Line2,** …: Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.
- **Counter0Start, Counter1Start, Counter2Start, …, Counter0End, Counter1End, Counter2End,** …: Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
- **Timer0Start, Timer1Start, Timer2Start, …, Timer0End, Timer1End, Timer2End,** …: Specifies which Timer signal to use as internal source for the transfer control trigger signal.
- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2**, …: Specifies which Software Signal to use as internal source for the transfer control trigger signal.
- **Action0, Action1, Action2,** …: Specifies which Action command to use as internal source for the transfer control trigger signal.

19.20 TransferTriggerActivation

Name	TransferTriggerActivation [TransferTriggerSelector]
Category	TransferControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Specifies the activation mode of the transfer control trigger.

Possible values are:

- **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. This can apply to TransferActive and TransferPause trigger.
- **LevelLow**: Specifies that the trigger is considered valid as long as the level of the source signal is low. This can apply to TransferActive and TransferPause trigger.

19.21 TransferStatusSelector

Name	TransferStatusSelector[TransferSelector]
Category	TransferControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	Streaming Paused Stopping Stopped QueueOverflow

Selects which status of the transfer module to read**.**

Possible values are:

- **Streaming**: Data blocks are transmitted when enough data is available.
- **Stopping:** Data blocks transmission is stopping. The current block transmission will be completed and the transfer state will stop**.**
- **Stopped**: Data blocks transmission is stopped.
- **Paused:** Data blocks transmission is suspended immediately.
- **QueueOverflow:** Data blocks queue is in overflow state.

19.22 TransferStatus

Name	TransferStatus[TransferStatusSelector]
Category	TransferControl
Level	Recommended
Interface	IBool
Access	Read

Unit	-
Visibility	Guru
Values	True False

Reads the status of the Transfer module signal selected by TransferStatusSelector**.**

19.23TransferComponentSelector

Name	TransferComponentSelector[TransferSelector]
Category	TransferControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Guru
Values	Red Green Blue All

Selects the color component for the control of the **TransferStreamChannel** feature.

Possible values are:

- **Red**: The **TransferStreamChannel** feature controls the index of the stream channel for the streaming of the red plane of the planar pixel formats.
- **Green**: The **TransferStreamChannel** feature controls the index of the stream channel for the streaming of the green plane of the planar pixel formats.
- **Blue**: The **TransferStreamChannel** feature controls the index of the stream channel for the streaming of blue plane of the planar pixel formats.
- **All:** The **TransferStreamChannel** feature controls the index of the stream channel for the streaming of all the planes of the planar pixel formats simultaneously or non planar pixel formats.

This feature is only needed if the device supports planar pixel formats.

19.24 TransferStreamChannel

Name	TransferStreamChannel[TransferSelector][Transfer

	ComponentSelector]
Category	TransferControl
Level	Optional
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Guru
Values	>0

Selects the streaming channel that will be used to transfer the selected stream of data. In general, this feature can be omitted and the default streaming channel will be used.

20 Sequencer Control

The Sequencer Control chapter describes the model and features related to the control of the sequencers that can be used to change some features of the device automaticaly based on different events and signals.

20.1 Sequencer control model

The Sequencer Control Model section describes the usage of the features related to the sequencing of automatic changes to features by the device. It describes the basic Sequencer model and the typical behavior of the device when a sequencer is used.

Sequencer overview

The purpose of a sequencer is to allow the user of a camera to define a series of feature sets for image acquisition which can consecutively be activated during the acquisition by the camera. Accordingly, the proposed sequence is configured by a list of parameter sets.

Each of these sequencer sets contains the settings for a number of camera features. Similar to user sets, the actual settings of the camera are overwritten when one of these sequencer sets is loaded. The order in which the features are applied to the camera depends on the design of the vendor. It is recommended to apply all the image related settings to the camera, before the first frame of this sequence is captured.

The sequencer sets can be loaded and saved by selecting them using **SequencerSetSelector**. The Execution of the sequencer is completely controlled by the device.

Configuration of a sequencer set

The index of the adjustable sequencer set is given by the **SequencerSetSelector**. The number of available sequencer sets is directly given by the range of this feature.

The features which are actually part of a sequencer set are defined by the camera manufacturer. These features can be read by **SequencerFeatureSelector** and activated by **SequencerFeatureEnable[SequencerFeatureSelector].** This configuration is the same for all Sequencer Sets.

To configure a sequencer set the camera has to be switched into configuration mode by **SequencerConfigurationMode**. Then the user has to select the desired sequencer set he wants to modify with the **SequencerSetSelector**. After the user has changed all the needed camera settings it is possible to store all these settings within a selected sequencer set by **SequencerSetSave[SequencerSetSelector].** The user can also read back this settings by **SequencerSetLoad[SequencerSetSelector]**.

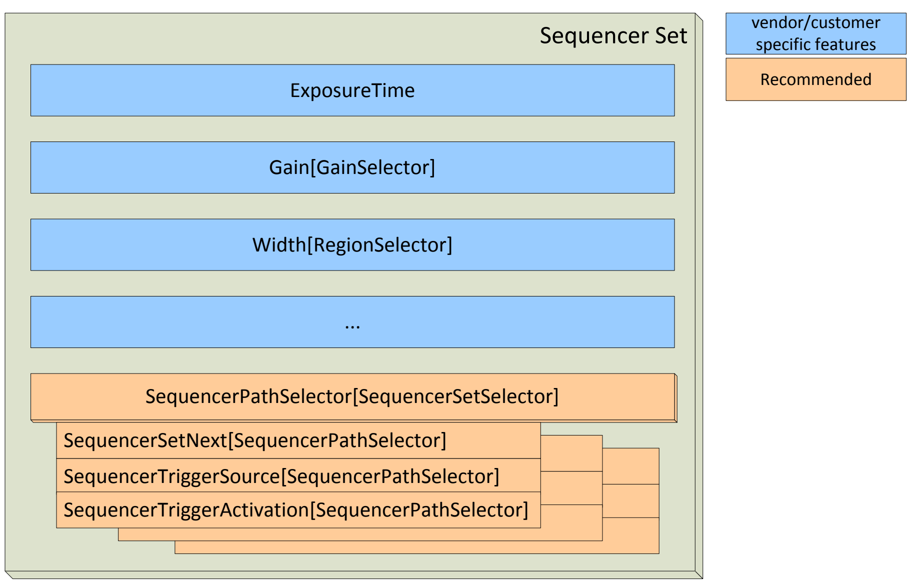
To permit a flexible usage, more than one possibility to go from one sequencer set to another can exist. Such a path is selected by **SequencerPathSelector[SequencerSetSelector]**. Each path and therefore the transition between different sequencer sets is based on a defined trigger and an aimed next sequencer set which is selectable by **SequencerSetNext[SequencerSetSelector][SequencerPathSelector].** After the trigger occurs the settings of the next set are active.

The trigger is defined by the features **SequencerTriggerSource[SequencerSetSelector][SequencerPathSelector]** and **SequencerTriggerActivation[SequencerSetSelector][SequencerPathSelector]**. The functions of these features are the same as **TriggerSource** and **TriggerActivation**. For a flexible sequencer implementation, the **SequencerPathSelector[SequencerSetSelector]** should be part of the sequencer sets.

A sequencer set should contain the following values:

- Camera data which should be controlled by the device
- **SequencerPathSelector[SequencerSetSelector]** with at least one path
- **SequencerSetNext**, **SequencerTriggerSource** and **SequencerTriggerActivation** for every path which is selectable by the **SequencerPathSelector.**

An example of a sequencer set is shown in the following figure:



Operation of a sequencer

The sequencer is started or stopped using the feature **SequencerMode**. If the sequencer is switched on, the start set which is defined by **SequencerSetStart** is loaded. The **SequencerStartSet** can take the same values as **SequencerSetSelector**.

While the sequencer is running, the **SequencerSetActive** is updated each time a new set is loaded. The feature can be used to read the current set – and the user might monitor the sequencer triggers.

If a trigger, which is selected by **SequencerTriggerSource[SequencerSetSelector][SequencerPathSelector]** and **SequencerTriggerActivation[SequencerSetSelector][SequencerPathSelector]** occurs, the sequencer switches to the next set. This set is configured by **SequencerSetNext[SequencerSetSelector][SequencerPathSelector]**, which is also part of a special sequencer path. If there is more than one path selected, the sequencer switches to the set whose trigger primarily occurred.

The matching between sequencer sets and currently taken frames can be realized with chunk mode or events.

20.2 Sequencer usage examples

Example 1: Simple features change while acquiring images

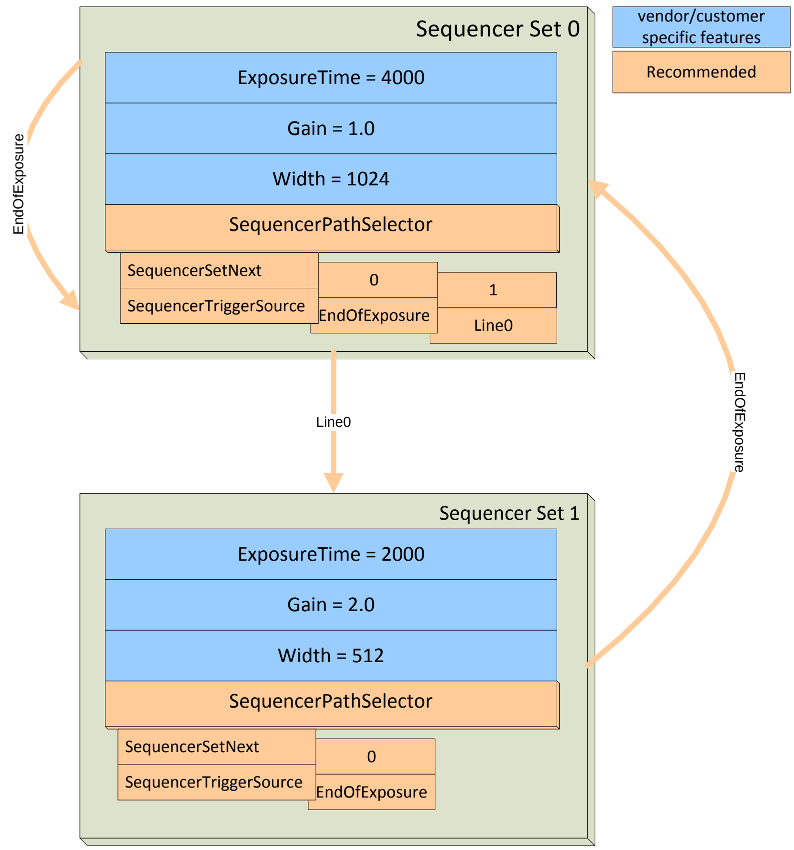
In this example, Set 0 and Set 1 are the main sets for the device (here a camera) to capture images. The only parameters within the sequencer sets are **Exposure Time, Gain, Width** and the (recommended) **SequencerPathSelector** and therefore **SequencerSetNext, SequencerTriggerSource**.

Set 0 is like a free running mode. After every image capture the camera stays in set 0. If a change on Line0 occurs, the sequencer switches to set 1 but only for 1 image capture. The parameters of the singles sets are the following:

- Set 0:
  - ExposureTime = 4000
  - Gain = 1.0
  - Width = 1024
  - SequencerSetNext[0] = 0
  - SequencerTriggerSource[0] = ExposureEnd
  - SequencerSetNext[1] = 1
  - SequencerTriggerSource[1] = Line0

- Set 1:
  - ExposureTime = 2000
  - Gain = 2.0
  - Width = 512
  - SequencerSetNext[0] = 0
  - SequencerTriggerSource[0] = ExposureEnd

The working diagram is shown in the following figure:



Example 2: Complex changes to features with multiples selectors.

In this example, the camera has 4 configurable sets. Set 0 and Set 1 are the main sets for the device to capture images. The only parameters within the sequencer sets are **Exposure Time, Gain** and the **SequencerPathSelector** and therefore **SequencerSetNext, SequencerTriggerSource.**

After the camera has captured an image with Set 0 the sequencer switches to Set 1. Then after the camera has captured another image with Set 1, the sequencer switches back to Set 0. So the most time the sequencer is alternating between Set 0 and Set 1. But 2 timers with different run times are also used to activate Set 2 and Set 3 when required.

The parameters of the individual sets are:

- **Set 0:**
  - ExposureTime = 4000
  - Gain = 1.0
  - SequencerSetNext[0] = 1
  - SequencerTriggerSource[0] = ExposureEnd
  - SequencerSetNext[1] = 3

- SequencerTriggerSource[1] = Timer0End

Set 1:

- ExposureTime = 2000
- Gain = 2.0
- SequencerSetNext[0] = 0
- SequencerTriggerSource[0] = ExposureEnd

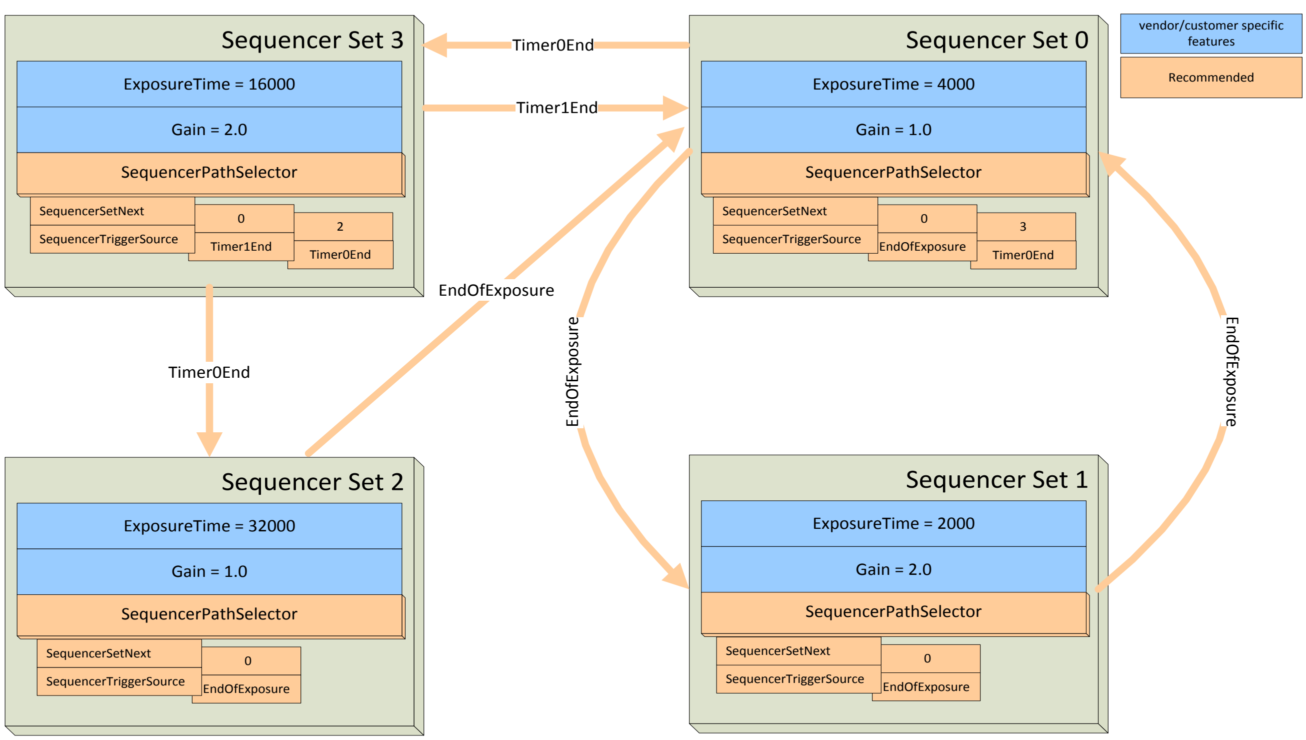
Set 2:

- ExposureTime = 32000
- Gain = 1.0
- SequencerSetNext[0] = 0
- SequencerTriggerSource[0] = EndOfExposure

Set 3:

- ExposureTime = 16000
- Gain = 2.0
- SequencerSetNext[0] = 0
- SequencerTriggerSource[0] = Timer1End
- SequencerSetNext[1] = 2
- SequencerTriggerSource[1] = Timer0End

The working diagram is shown in the following figure:



20.3 Sequencer Control features

This section describes in detail the features related to the Sequencer Control.

20.4 SequencerControl

Name	SequencerControl
Category	Root
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Expert
Values	-

Category for the Sequencer Control features.

20.5 SequencerMode

Name	SequencerMode
Category	SequencerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	On Off

Controls if the sequencer mechanism is active.

Possible values are:

- **Off:** Disables the sequencer.
- **On:** Enables the sequencer.

20.6 SequencerConfigurationMode

Name	SequencerConfigurationMode
Category	SequencerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	On Off

Controls if the sequencer configuration mode is active.

Possible values are:

- **Off:** Disables the sequencer configuration mode.
- **On:** Enables the sequencer configuration mode.

20.7 SequencerFeatureSelector

Name	SequencerFeatureSelector
Category	SequencerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Device-Specific

Selects which sequencer features to control.

The feature lists all the features that can be part of a device sequencer set. All the device's sequencer sets have the same features.

Note that the name used in the enumeration must match exactly the device's feature name.

20.8 SequencerFeatureEnable

Name	SequencerFeatureEnable[SequencerFeatureSelector]

Category	SequencerControl
Level	Recommended
Interface	IBoolean
Access	Read/(Write)
Unit	-
Visibility	Expert
Values	True False

Enables the selected feature and make it active in all the sequencer sets.

20.9 SequencerSetSelector

Name	SequencerSetSelector
Category	SequencerControl
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Selects the sequencer set to which further feature settings applies.

20.10 SequencerSetSave

Name	SequencerSetSave[SequencerSetSelector]
Category	SequencerControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Saves the current device state to the sequencer set selected by the **SequencerSetSelector**.

20.11 SequencerSetLoad

Name	SequencerSetLoad[SequencerSetSelector]
Category	SequencerControl
Level	Recommended
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Expert
Values	-

Loads the sequencer set selected by **SequencerSetSelector** in the device. Even if **SequencerMode** is off, this will change the device state to the configuration of the selected set.

20.12 SequencerSetActive

Name	SequencerSetActive
Category	SequencerControl
Level	Optional
Interface	IInteger
Access	Read
Unit	-
Visibility	Expert
Values	≥0

Contains the currently active sequencer set.

20.13 SequencerSetStart

Name	SequencerSetStart
Category	SequencerControl
Level	Recommended
Interface	IInteger

Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Sets the initial/start sequencer set, which is the first set used within a sequencer.

20.14 SequencerPathSelector

Name	SequencerPathSelector[SequencerSetSelector]
Category	SequencerControl
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Selects to which branching path further path settings applies.

20.15 SequencerSetNext

Name	SequencerSetNext[SequencerSetSelector] [SequencerPathSelector]
Category	SequencerControl
Level	Recommended
Interface	IInteger
Access	Read/Write
Unit	-
Visibility	Expert
Values	≥0

Specifies the next sequencer set.

20.16 SequencerTriggerSource

Name	SequencerTriggerSource[SequencerSetSelector] [SequencerPathSelector]
Category	SequencerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Expert
Values	Off AcquisitionTrigger AcquisitionStart AcquisitionEnd FrameTrigger FrameStart FrameEnd FrameBurstStart FrameBurstEnd ExposureStart ExposureEnd Line0 (If 0 based), Line1, Line2, … UserOutput0, UserOutput1, UserOutput2, … Counter0Start (If 0 based), Counter1Start, Counter2Start, … Counter0End (If 0 based), Counter1End, Counter2End, … Timer0Start (If 0 based), Timer1Start, Timer2Start, … Timer0End (If 0 based), Timer1End, Timer2End, … SoftwareSignal0 (If 0 based), SoftwareSignal1, SoftwareSignal2, … Action0 (If 0 based), Action1, Action2, … CC1, CC2, CC3, CC4

Specifies the internal signal or physical input line to use as the sequencer trigger source.

Possible values are:

- **Off**: Disables the sequencer 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 Start Trigger.
- **FrameStart**: Starts with the reception of the Frame Start.
- **FrameEnd**: Starts with the reception of the Frame End.
- **FrameBurstStart**: Starts with the reception of the Frame Burst Start.
- **FrameBurstEnd**: Starts with the reception of the Frame Burst End.
- **ExposureStart**: Starts with the reception of the Exposure Start.
- **ExposureEnd**: Starts with the reception of the Exposure End.
- **Line0** (If 0 based)**, Line1, Line2, …**: Starts when the specidfied TimerTriggerActivation condition is met on the chosen I/O Line.
- **UserOutput0, UserOutput1, UserOutput2, …**: Specifies which User Output bit signal to use as internal source for the trigger.
- **Counter0Start, Counter1Start, Counter2Start, …**: Starts with the reception of the Counter Start.
- **Counter0End, Counter1End, Counter2End, …**: Starts with the reception of the Counter End.
- **Timer0Start, Timer1Start, Timer2Start, …**: Starts with the reception of the Timer Start.
- **Timer0End, Timer1End, Timer2End, …**: Starts with the reception of the Timer End.
- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2**, **…**: Starts on the reception of the Software Signal.
- **Action0, Action1, Action2, …:** Starts with the assertion of the chosen action signal.
- **CC1, CC2, CC3, CC4**: Index of the Camera Link physical line and associated I/O control block to use. This ensures a direct mapping between the lines on the frame grabber and on the camera. Applicable to CameraLink products only.

20.17 SequencerTriggerActivation

Name	SequencerTriggerActivation[SequencerSetSelector][SequencerPathSelector]
Category	SequencerControl
Level	Recommended
Interface	IEnumeration
Access	Read/Write
Unit	-

Visibility	Expert
Values	RisingEdge FallingEdge AnyEdge LevelHigh LevelLow

Specifies the activation mode of the sequencer trigger.

Possible values are:

- **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.

21 Software Signal Control

The Software Signal Control chapter describes the model and features related to the control and the generation of software generated signals.

A software signal is a general source and can be used as a trigger signal source for diverse functions in other SFNC modules.

21.1 SoftwareSignalControl

Name	SoftwareSignalControl
Category	Root
Level	Optional
Interface	ICategory
Access	Read
Unit	-
Visibility	Beginner
Values	-

Category that contains the Software Signal Control features.

21.2 SoftwareSignalSelector

Name	SoftwareSignalSelector
Category	SoftwareSignalControl
Level	Optional
Interface	IEnumeration
Access	Read/Write
Unit	-
Visibility	Beginner
Values	SoftwareSignal0 (If 0 based) SoftwareSignal1, SoftwareSignal2, …

Selects which Software Signal features to control.

Possible values are:

- **SoftwareSignal0, SoftwareSignal1, SoftwareSignal2, …**: Selects the software generated signal to control.

21.3 SoftwareSignalPulse

Name	SoftwareSignalPulse[SoftwareSignalSelector]
Category	SoftwareSignalControl
Level	Optional
Interface	ICommand
Access	(Read)/Write
Unit	-
Visibility	Beginner
Values	-

Generates a pulse signal that can be used as a software trigger. This command can be used to trigger other modules that accept a SoftwareSignal as trigger source.

22 Typical Standard Feature Usage Examples

This chapter shows examples of typical use cases of the standard acquisition and control of the SFNC 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).

22.1 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-Frame 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;
TriggerDelay          = 1000;
ExposureMode          = Timed;
ExposureTime          = 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. An exposure end event is also sent to the Host application after the exposure of each frame to signal that the inspected part can be moved. The timestamp of the event is also read.
*/

```
AcquisitionMode    = Continuous;
TriggerSelector    = FrameStart;
TriggerMode        = On;
TriggerActivation  = RisingEdge;
TriggerSource      = Line2;
ExposureMode       = Timed;
ExposureTime       = 500;
Register(Camera.EventExposureEnd, CallbackDataObject, CallbackFunctionPtr)
EventSelector      = ExposureEnd;
EventNotification  = On;
AcquisitionStart();
…
// In the callback of the ExposureEnd event, get the event timestamp:
Timestamp = EventExposureEndTimestamp;
…
AcquisitionStop();
```

```
/* Multi-Frame 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 event.
*/
```

```
AcquisitionMode        = MultiFrame;
AcquisitionFrameCount = 20;
TriggerSelector        = FrameStart;
TriggerMode            = On;
TriggerActivation      = RisingEdge;
TriggerSource          = Line1;
ExposureMode           = Timed;
ExposureTime           = 500;
LineSelector           = Line2;
LineMode               = Output;
LineInverter           = True;
LineSource             = ExposureActive
Register(Camera.EventAcquisitionEnd,CallbackDataObject,CallbackFunctionPtr)
EventSelector          = AcquisitionEnd;
EventNotification      = On;
AcquisitionStart();
```

/* Continuous Acquisition of frames in bursts of 10 frames. Each burst is triggered by a Hardware trigger on Line 1. The end of each burst capture is signalled to the host with a FrameBurstEnd event.
*/

```
AcquisitionMode       = Continuous;
AcquisitionBurstFrameCount = 10;
TriggerSelector       = FrameBurstStart;
TriggerMode           = On;
TriggerActivation     = RisingEdge;
TriggerSource         = Line1;
Register(Camera.EventFrameBurstEnd,CallbackDataObject,CallbackFunctionPtr)
EventSelector         = FrameBurstEnd;
EventNotification     = On;
AcquisitionStart();
…
// In the callback of the end of burst event, get the event timestamp:
Timestamp = EventExposureEndTimestamp;
…
AcquisitionStop();
```

/* Multi-Frame Acquisition of 50 frames in 5 bursts of 10 frames. Each burst is triggered by a Hardware trigger on Line 1.
*/

```
AcquisitionMode       = MultiFrame;
AcquisitionFrameCount = 50;
AcquisitionBurstFrameCount = 10;
TriggerSelector       = FrameBurstStart;
TriggerMode           = On;
TriggerActivation     = RisingEdge;
TriggerSource         = Line1;
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 = AcquisitionStart;
TriggerMode     = On;
TriggerSource   = Line1;
ExposureMode    = TriggerControlled;
TriggerSelector = ExposureStart;
TriggerMode     = On;
TriggerSource   = Line3;
TriggerSelector = ExposureStop;
TriggerMode     = On;
TriggerSource   = Line4;
AcquisitionStart();
...
AcquisitionStop();
```

22.2 Counter and Timer Examples

/* Counts the number of rising edge triggers signals received on Line1 and the number of Frame Start events in Hardware triggered Continuous acquisition to verify that no trigger were missed.
*/

```
AcquisitionMode    = Continuous;
TriggerSelector    = FrameStart;
TriggerActivation  = RisingEdge;
TriggerSource      = Line1;
TriggerMode        = On;

CounterSelector    = Counter1;
CounterEventSource = Line1;
CounterEventActivation = RisingEdge;
CounterReset();

CounterSelector    = Counter2;
CounterEventSource = FrameStart;
CounterReset();

AcquisitionStart();
…
AcquisitionStop();

CounterSelector    = Counter1;
NbTrigger          = CounterValue;
CounterSelector    = Counter2;
NbFrameStart       = CounterValue;
if (NbTrigger != NbFrameStart)
    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;
CounterResetSource    = CounterTrigger;
Register(Camera.EventCounter1End,CallbackDataObject,CallbackFunctionPtr)
EventSelector         = Counter1End;
EventNotification     = On;
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;
TimerDuration          = 200;
TimerDelay             = 100;
TimerTriggerSource     = Line1;
TimerTriggerActivation = RisingEdge;
LineSelector           = Line2;
LineMode               = Output;
LineSource             = Timer1Active;
```

/* 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;
Register(Camera.EventLine1RisingEdge,CallbackDataObject,CallbackFunctionPtr)
EventSelector          = Line1RisingEdge;
EventNotifications     = On;
/* Wait for the event on the host to read the time. */
…
TimerSelector          = Timer1;
PulseDuration          = TimerValue;
```

22.3 I/O Examples

/* Read the inverted Status of the physical Line 1. */

```
LineSelector  = Line1;
LineMode      = Input;
LineInverter  = True;
CurrentStatus  = 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;
```

22.4 Action Signal Examples

/* Triggered Single Frame acquisition using the Action Signal 1. */

```
AcquisitionMode     = SingleFrame;
TriggerSelector     = FrameStart;
TriggerMode         = On;
TriggerSource       = Action1;

ActionDeviceKey     = 0x12345678;
ActionSelector      = 1
ActionGroupKey      = 0x1
ActionGroupMask     = 0x1

AcquisitionStart();
```

/* Generate a 200us Timer pulse (Strobe) on the physical output Line 2.
The Timer pulse is started using a trigger coming from Action Signal 3.
*/

```
TimerSelector           = Timer1;
TimerDuration           = 200;
TimerTriggerSource      = Action3;

LineSelector            = Line2;
LineMode                = Output;
LineSource              = Timer1Active;

ActionDeviceKey         = 0x12345678;
ActionSelector          = Action3
ActionGroupKey          = 0x1
ActionGroupMask         = 0x7
```

23 Acknowledgements

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

Company	Represented by
Teledyne DALSA	Eric Carey, Eric Bourbonnais, Peifang Zhou
SVS-Vistek	Gerd Reichle
Stemmer Imaging	Sascha Dorenbeck, Rupert Stelz, Martin Kersting
SICK IVP	Mattias Johannesson
Pleora Technologies	Francois Gobeil, Vincent Rowley, Geoff Roddick
National Instruments	Chris Graf, Tony Iglesias, Johann Scholtz, Eric Gross
MVTec Software	Christoph Zierl
Matrox Imaging	Stéphane Maurice
MathWorks	Mark Jones
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
Basler	Fritz Dierks, Alex Happe, Andreas Gaer, Thies Möller, Sven Seeger
Baumer	Marcel Naggatz
AIA	Jeff Fryman

24 Tap Geometry Appendix

24.1 Motivations

This appendix defines standard names to uniquely identify the geometrical properties of most device`s sensor taps.

For example, the initial release of the Camera Link® standard didn`t include any information about the geometrical properties of taps.

Frame grabbers are able to reconstruct the image from multi-tap cameras on-the-fly.

Camera manufacturers should clearly specify what geometry(ies) is (are) required. Frame grabber manufacturers should also clearly specify what geometry(ies) is (are) supported.

The customer can then check the compatibility, and select the appropriate geometry for the camera and the frame grabber.

Considering the limited amount of cases, a unique name is assigned for each geometry.

24.2 Identifying the Geometrical Properties

24.2.1 Image Geometrical Properties

The relevant geometrical properties required for reconstructing the image:

- **Vantage point**: An enumerated value that specifies the position of the pixel with coordinate X=1, Y=1 in the scene.
  {Top-Left, Top-Right, Bottom-Left, Bottom-Right}
  Default is Top-Left.
- **ImageWidth**: An integer value declaring the image width expressed in pixels.
- **ImageHeight**: An integer value declaring the image height expressed in pixels. This parameter is irrelevant in case of line-scan or TDI cameras.
- **TapGeometry**: An enumerated type of parameter that summarizes the following properties for each tap:
  - XStart: X-coordinate of the first pixel column
  - YStart: Y-coordinate if the first pixel row
  - XEnd: X-coordinate of the last pixel column
  - YEnd: Y-coordinate of the last pixel row
  - XStep: Difference of X-coordinates between consecutive pixel columns; X-step is positive when X-coordinates are increasing along a row; it is negative otherwise.

- YStep: Difference of Y-coordinates between consecutive pixel rows; Y-step is positive when Y-coordinates is increasing at the end of a line; it is negative otherwise.
- Allocation of taps to ports. The camera taps are indexed using following conventional sorting rule:
  First by increasing values of YStart then by increasing value of XStart. The tap T1 is the sensor tap that exhibits the smallest XStart for the smallest YStart.

24.2.1.1 Restrictions

- All zones have the same size.
- Zones do not overlap.
- All zones have the same number of taps.
- All taps are carrying the same amount of pixels.

24.2.1.2 Tap Naming Convention

A tap configuration for **area-scan** camera is designated by:

<TapGeometryX>**-**<TapGeometryY>

A tap configuration for **line-scan or TDI-line-scan** camera is designated by:

<TapGeometryX>

TapGeometryX is designated by <ZoneX>**X**(<TapX>)(<ExtX>)>)

<ZoneX>: An integer in the range of {1, 2, 3, 4, 8, 10} declaring number of zones encountered across horizontal direction.

<TapX>: An integer in range {∅, 2, 3, 4, 8, 10} declaring the number of consecutive pixels in the horizontal direction that are outputted simultaneously from a zone. This field is omitted when all pixels are in the same column.

<ExtX>: A letter in the range of {∅, E, M} declaring the location of the pixels extractors in the horizontal direction. The value E indicates that pixel extractors are at both ends of the line. Value M indicates that pixel extractors are in the middle of the line. This field is omitted when all pixel extractors are all at the left end of each zone.

TapGeometryY is designated by <ZoneY>Y(<TapY>)(<ExtY>)>)

<ZoneY>: An integer in the range of {1, 2} declaring the number of zones encountered in the vertical direction.

<TapY>: An integer in the range iof {∅, 2} declaring the number of consecutive pixels in vertical direction that are outputted simultaneously from each zone. This field is omitted when all pixels are in the same line.

<ExtY>: A letter in range of {∅, E} declaring the location of the pixels extractors in the vertical direction. The value E indicates that pixel extractors are at both top and bottom lines. This field is omitted when all pixel extractors are in the top line.

24.2.1.3 Tap Geometrical Properties

The following tables provide description of all the tap geometry configurations. For every configuration the first and last pixel belonging to that tap, as well as the pixel increment corresponding to the given tap is listed.

This table enumerates the standard tap geometries. The table is sorted by increasing number of taps. It displays the values of the 6 geometrical properties for each tap.

Table 24-1 Tap geometrical properties – One, two and three taps

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
1X-1Y	1X	Tap1	1	W	1	1	H	1
1X2-1Y	1X2	Tap1	1	W-1	2	1	H	1
		Tap2	2	W	2	1	H	1
2X-1Y	2X	Tap1	1	W/2	1	1	H	1
		Tap2	W/2+1	W	1	1	H	1
2XE-1Y	2XE	Tap1	1	W/2	1	1	H	1
		Tap2	W	W/2+1	-1	1	H	1
2XM-1Y	2XM	Tap1	W/2	1	-1	1	H	1
		Tap2	W/2+1	W	1	1	H	1
1X-1Y2		Tap1	1	W	1	1	H-1	2
		Tap2	1	W	1	2	H	2
1X-2YE		Tap1	1	W	1	1	H/2	1
		Tap2	1	W	1	H	H/2+1	-1
1X3-1Y	1X3	Tap1	1	W-2	3	1	H	1
		Tap2	2	W-1	3	1	H	1
		Tap3	3	W	3	1	H	1
3X-1Y	3X	Tap1	1	W/3	1	1	H	1

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
		Tap2	W/3+1	2W/3	1	1	H	1
		Tap3	2W/3+1	W	1	1	H	1

Table 24-2 Tap geometrical properties – Four taps

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
1X4-1Y	1X4	Tap1	1	W-3	4	1	H	1
		Tap2	2	W-2	4	1	H	1
		Tap3	3	W-1	4	1	H	1
		Tap4	4	W	4	1	H	1
4X-1Y	4X	Tap1	1	W/4	1	1	H	1
		Tap2	W/4+1	W/2	1	1	H	1
		Tap3	W/2+1	3W/4	1	1	H	1
		Tap4	3W/4+1	W	1	1	H	1
2X2-1Y	2X2	Tap1	1	W/2-1	2	1	H	1
		Tap2	2	W/2	2	1	H	1
		Tap3	W/2+1	W-1	2	1	H	1
		Tap4	W/2+2	W	2	1	H	1
2X2E-1Y	2X2E	Tap1	1	W/2-1	2	1	H	1
		Tap2	2	W/2	2	1	H	1
		Tap3	W-1	W/2+1	-2	1	H	1
		Tap4	W	W/2+2	-2	1	H	1
2X2M-1Y	2X2M	Tap1	W/2-1	1	-2	1	H	1
		Tap2	W/2	2	-2	1	H	1
		Tap3	W/2+1	W-1	2	1	H	1
		Tap4	W/2+2	W	2	1	H	1
1X2-2YE		Tap1	1	W-1	2	1	H/2	1
		Tap2	2	W	2	1	H/2	1
		Tap3	1	W-1	2	H	H/2+1	-1
		Tap4	2	W	2	H	H/2+1	-1
2X-2YE		Tap1	1	W/2	1	1	H/2	1
		Tap2	W/2+1	W	1	1	H/2	1
		Tap3	1	W/2	1	H	H/2+1	-1
		Tap4	W/2+1	W	1	H	H/2+1	-1
2XE-2YE		Tap1	1	W/2	1	1	H/2	1
		Tap2	W	W/2+1	-1	1	H/2	1
		Tap3	1	W/2	1	H	H/2+1	-1
		Tap4	W	W/2+1	-1	H	H/2+1	-1
2XM-2YE		Tap1	W/2	1	-1	1	H/2	1
		Tap2	W/2+1	W	1	1	H/2	1
		Tap3	W/2	1	-1	H	H/2+1	-1
		Tap4	W/2+1	W	1	H	H/2+1	-1

Table 24-3 Tap geometrical properties –Eight taps

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
1X8-1Y	1X8	Tap1	1	W-7	8	1	H	1
		Tap2	2	W-6	8	1	H	1
		Tap3	3	W-5	8	1	H	1
		Tap4	4	W-4	8	1	H	1
		Tap5	5	W-3	8	1	H	1
		Tap6	6	W-2	8	1	H	1
		Tap7	7	W-1	8	1	H	1
		Tap8	8	W	8	1	H	1
8X-1Y	8X	Tap1	1	W/8	1	1	H	1
		Tap2	W/8+1	2W/8	1	1	H	1
		Tap3	2W/8+1	3W/8	1	1	H	1
		Tap4	3W/8+1	4W/8	1	1	H	1
		Tap5	4W/8+1	5W/8	1	1	H	1
		Tap6	5W/8+1	6W/8	1	1	H	1
		Tap7	6W/8+1	7W/8	1	1	H	1
		Tap8	7W/8+1	W	1	1	H	1
4X2-1Y	4X2	Tap1	1	W/4-1	2	1	H	1
		Tap2	2	W/4	2	1	H	1
		Tap3	W/4+1	W/2-1	2	1	H	1
		Tap4	W/4+1	W/2	2	1	H	1
		Tap5	W/2+1	3W/4-1	2	1	H	1
		Tap6	W/2+2	3W/4	2	1	H	1
		Tap7	3W/4+1	W-1	2	1	H	1
		Tap8	3W/4+2	W	2	1	H	1
4X2E-1Y	4X2E	Tap1	1	W/4-1	2	1	H	1
		Tap2	2	W/4	2	1	H	1
		Tap3	W/4+1	W/2-1	2	1	H	1
		Tap4	W/4+2	W/2	2	1	H	1
		Tap5	3W/4-1	W/2+1	-2	1	H	1
		Tap6	3W/4	W/2+2	-2	1	H	1
		Tap7	W-1	3W/4+1	-2	1	H	1
		Tap8	W	3W/4+2	-2	1	H	1
2X2E-2YE		Tap1	1	W/2-1	2	1	H/2	1
		Tap2	2	W/2	2	1	H/2	1
		Tap3	W-1	W/2+1	-2	1	H/2	1
		Tap4	W	W/2+2	-2	1	H/2	1
		Tap5	1	W/2-1	2	H	H/2+1	-1
		Tap6	2	W/2	2	H	H/2+1	-1

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
		Tap7	W-1	W/2+1	-2	H	H/2+1	-1
		Tap8	W	W/2+2	-2	H	H/2+1	-1

Table 24-4 Tap geometrical properties –Ten taps

Geometry name		Tap	Tap geometrical properties					
Area-scan	Line-scan		X Start	X End	Step X	Y Start	Y End	Step Y
1X10-1Y	1X10	Tap1	1	W-9	10	1	H	1
		Tap2	2	W-8	10	1	H	1
		Tap3	3	W-7	10	1	H	1
		Tap4	4	W-6	10	1	H	1
		Tap5	5	W-5	10	1	H	1
		Tap6	6	W-4	10	1	H	1
		Tap7	7	W-3	10	1	H	1
		Tap8	8	W-2	10	1	H	1
		Tap9	9	W-1	10	1	H	1
		Tap10	10	W	10	1	H	1
10X-1Y	10X	Tap1	1	W/10	1	1	H	1
		Tap2	W/10+1	2W/10	1	1	H	1
		Tap3	2W/10+1	3W/10	1	1	H	1
		Tap4	3W/10+1	4W/10	1	1	H	1
		Tap5	4W/10+1	5W/10	1	1	H	1
		Tap6	5W/10+1	6W/10	1	1	H	1
		Tap7	6W/10+1	7W/10	1	1	H	1
		Tap8	7W/10+1	8W/10	1	1	H	1
		Tap9	8W/10+1	9W/10	1	1	H	1
		Tap10	9W/10+1	W	1	1	H	1

24.3 Tap Geometry Drawings

24.3.1 Single Tap Geometry

1X-1Y (area-scan)

1 zone in X, 1 zone in Y.

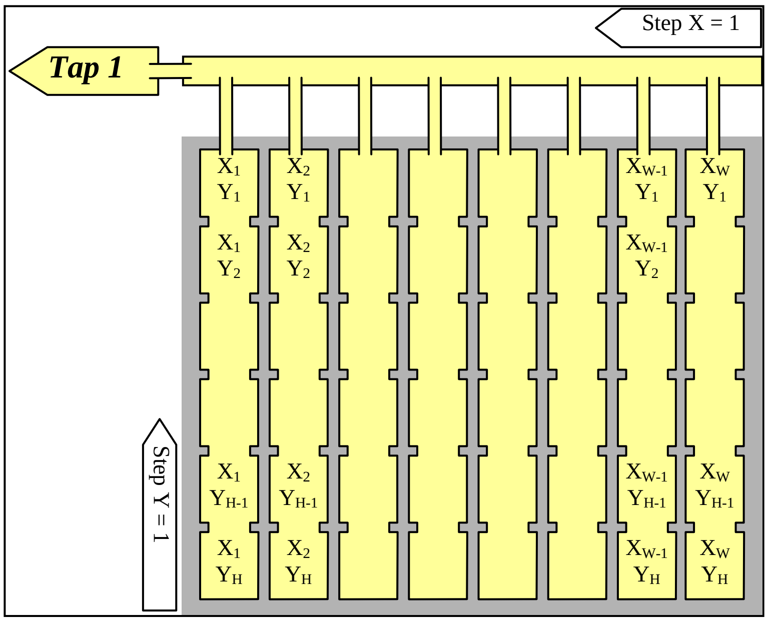


Figure 24-1 Geometry 1X-1Y (area-scan)

1X (line-scan)

1 zone in X.

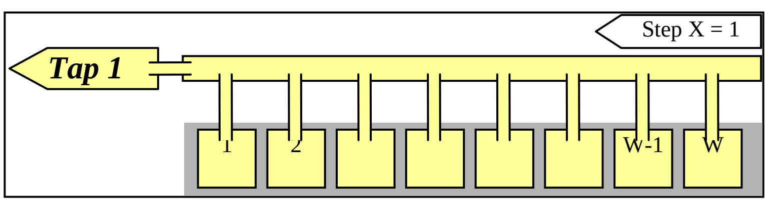


Figure 24-2 Geometry 1X (line-scan)

24.3.2 Dual Tap Geometries

1X2-1Y (area-scan)

1 zone in X with 2 taps, 1 zone in Y.

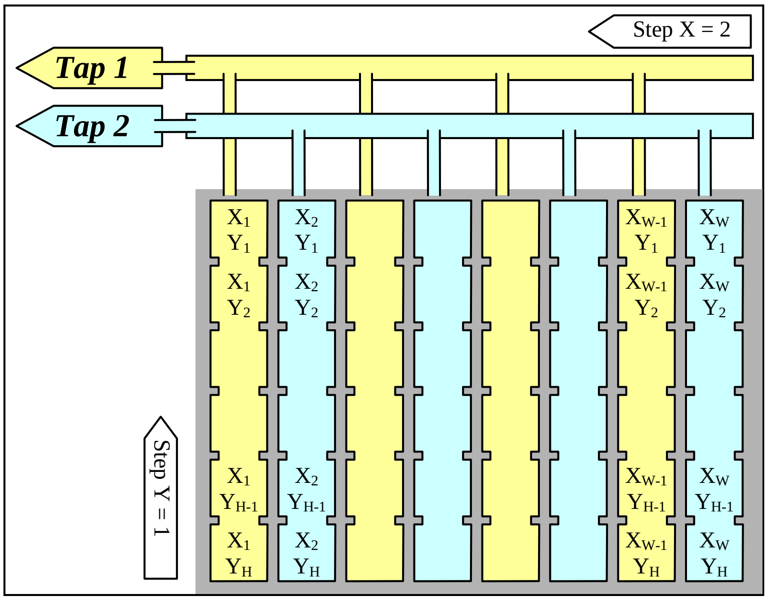


Figure 24-3 Geometry 1X2-1Y (area-scan)

1X2 (line-scan)

1 zone in X with 2 taps.

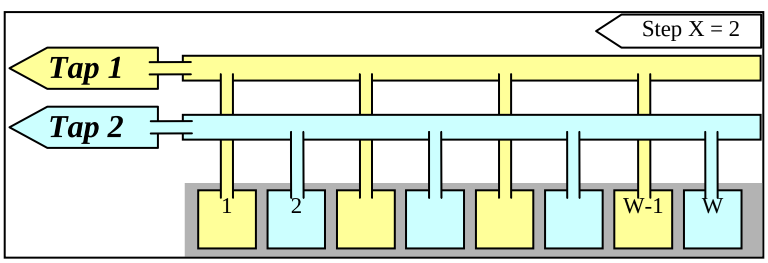


Figure 24-4 Geometry 1X2 (line-scan)

2X-1Y (area-scan)

2 zones in X, 1 zone in Y.

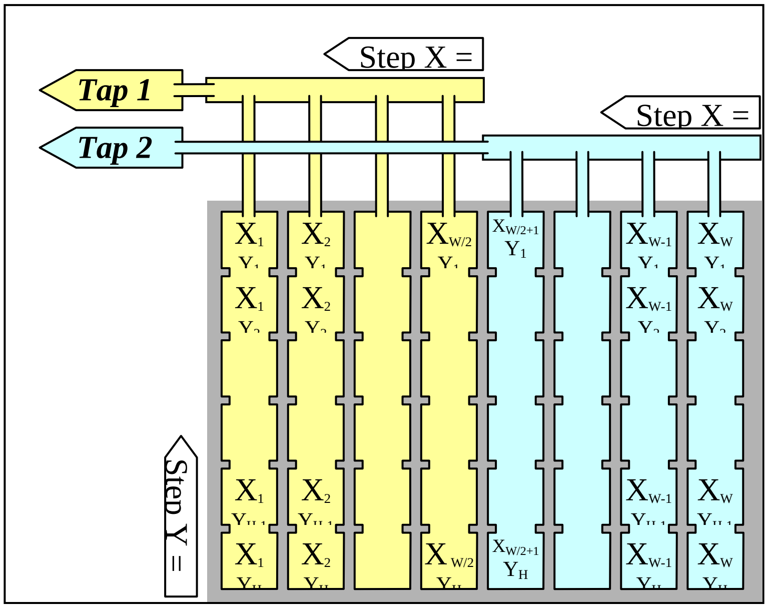


Figure 24-5 Geometry 2X-1Y (area-scan)

2X (line-scan)

2 zones in X.

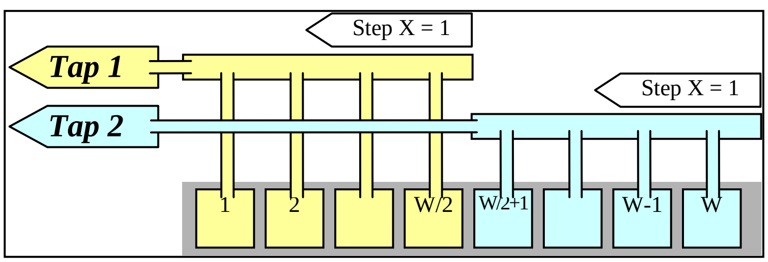


Figure 24-6 Geometry 2X (line-scan)

2XE-1Y (area-scan)

2 zones in X with end extraction, 1 zone in Y.

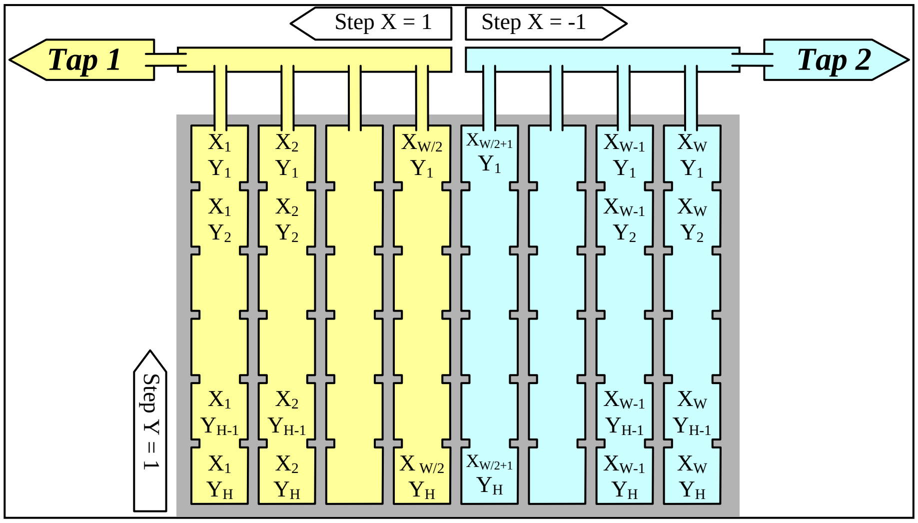


Figure 24-7 Geometry 2XE-1Y (area-scan)

2XE (line-scan)

2 zones in X with end extraction.

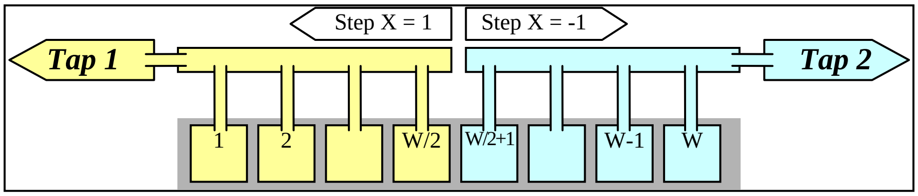


Figure 24-8 Geometry 2XE (line-scan)

2XM-1Y (area-scan)

2 zones in X with middle extraction, 1 zone in Y.

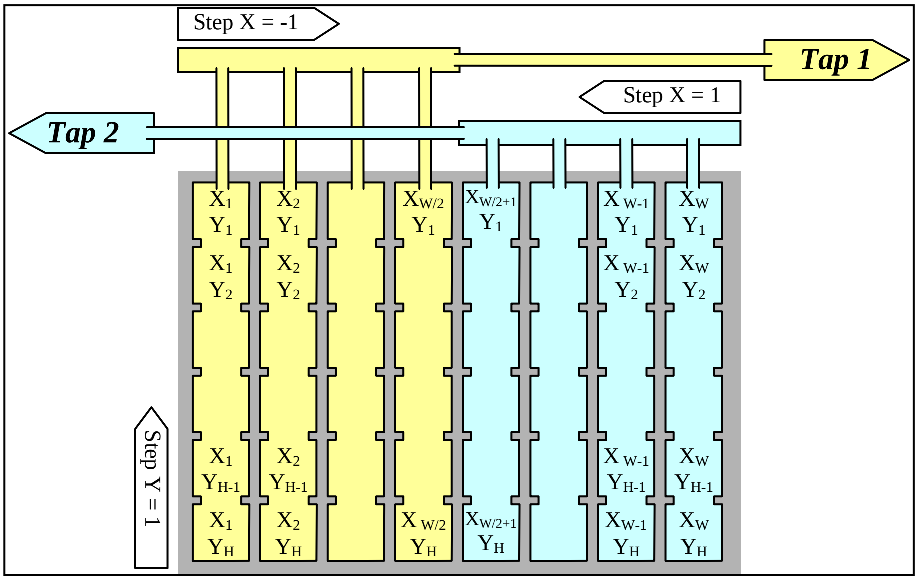


Figure 24-9 Geometry 2XM-1Y (area-scan)

2XM (line-scan)

2 zones in X with middle extraction.

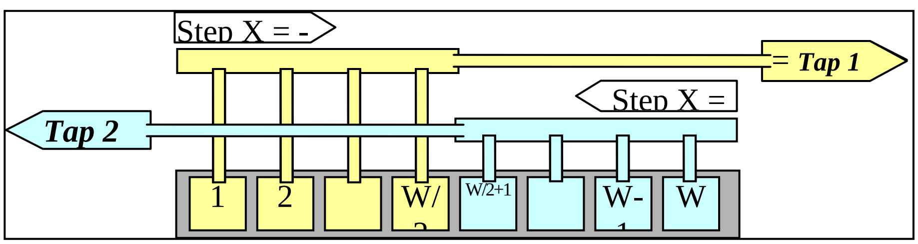


Figure 24-10 Geometry 2XM (line-scan)

1X-1Y2 (area-scan)

1 zone in X, 1 zone in Y with 2 taps.

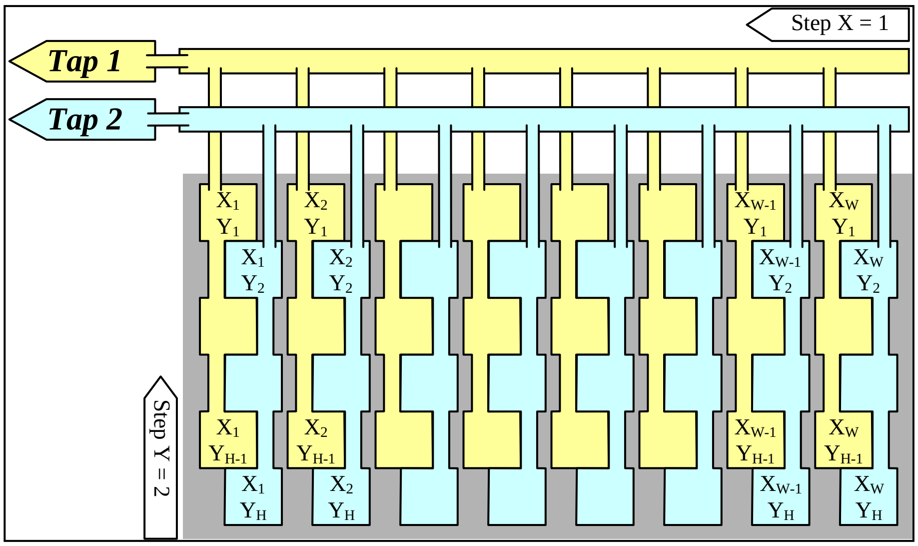


Figure 24-11 Geometry 1X-1Y2 (area-scan)

1X-2YE (area-scan)

1 zone in X, 2 zones in Y with end extraction.

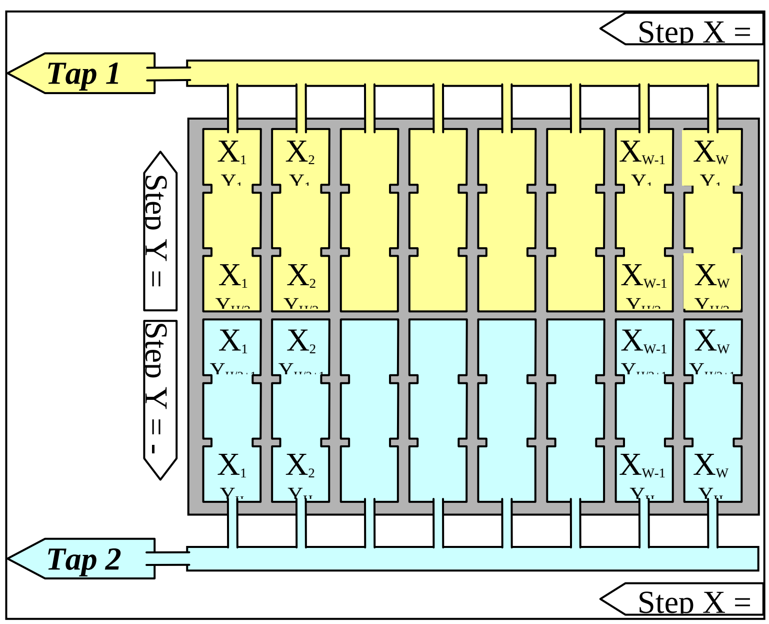


Figure 24-12 Geometry 1X-2YE (area-scan)

24.4 Triple Tap Geometries

1X3-1Y (area-scan)

1 zone in X with 3 taps, 1 zone in Y.

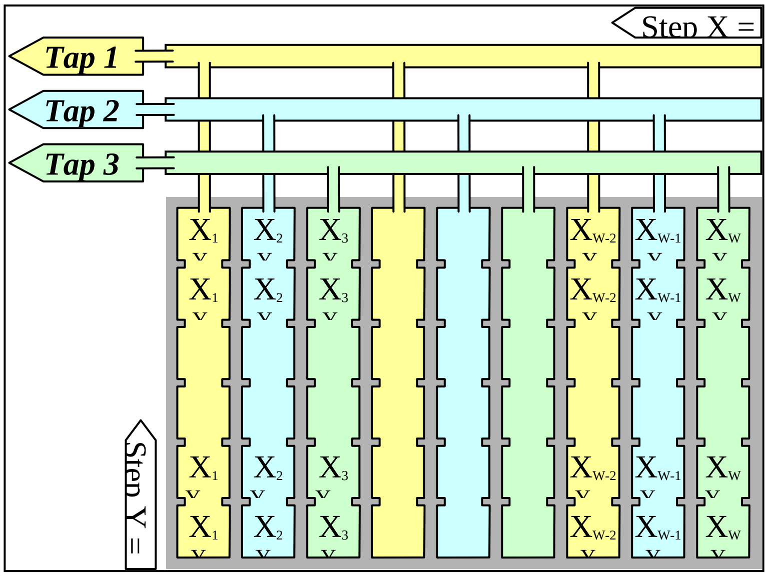


Figure 24-13 Geometry 1X3-1Y (area-scan)

1X3 (line-scan)

1 zone in X with 3 taps.

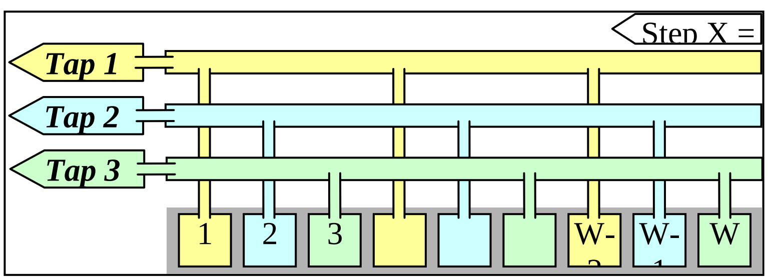


Figure 24-14 Geometry 1X3 (line-scan)

3X-1Y (area-scan)

3 zones in X, 1 zone in Y.

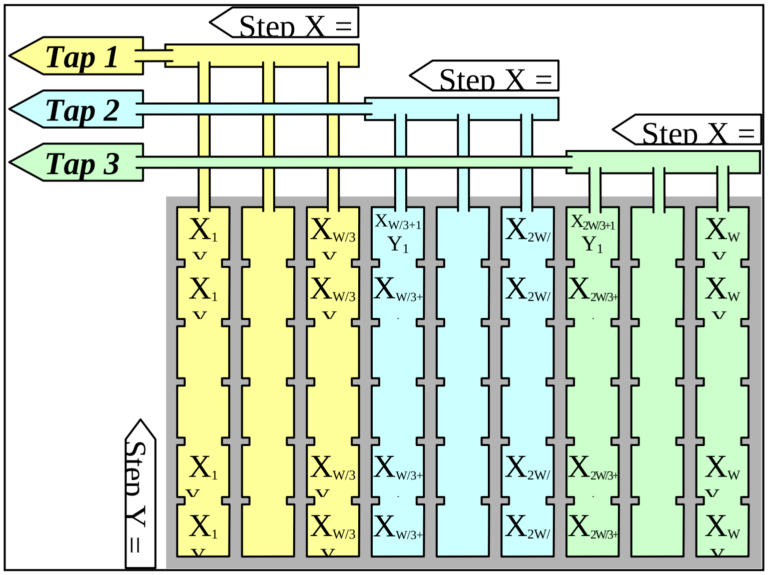


Figure 24-15 Geometry 1X3-1Y (area-scan)

3X (line-scan)

3 zones in X.

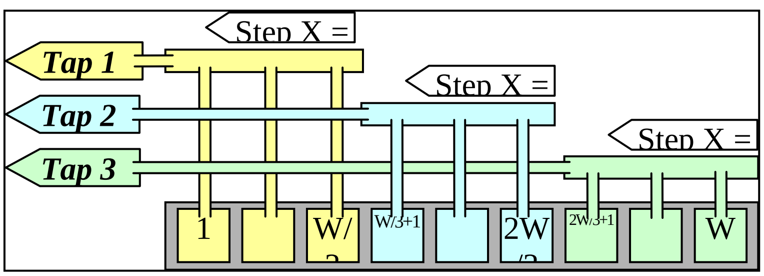


Figure 24-16 Geometry 1X3 (line-scan)

24.5 Quad Tap Geometries

1X4-1Y (area-scan)

1 zone in X with 4 taps, 1 zone in Y.

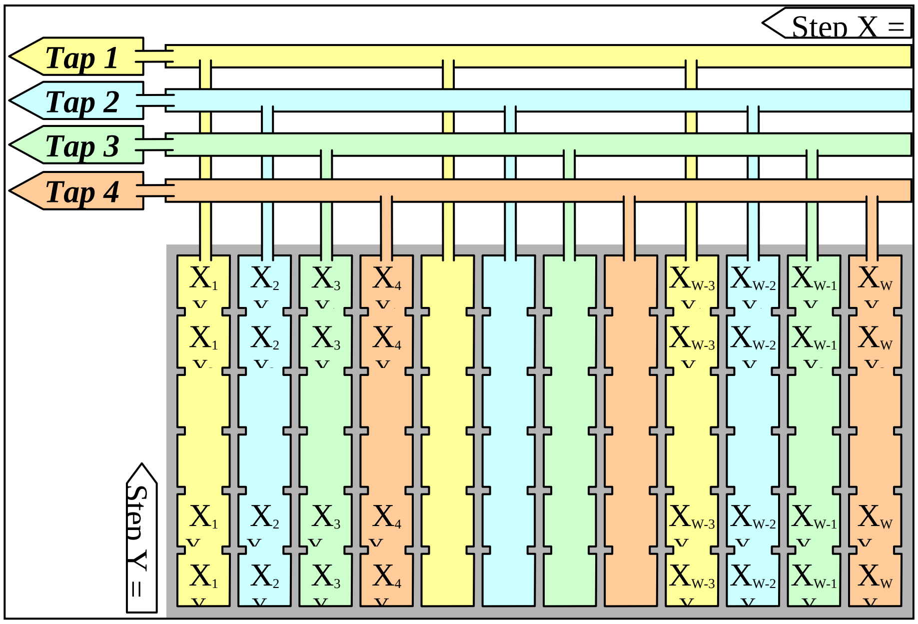


Figure 24-17 Geometry 1X4-1Y (area-scan)

1X4 (line-scan)

1 zone in X with 4 taps.

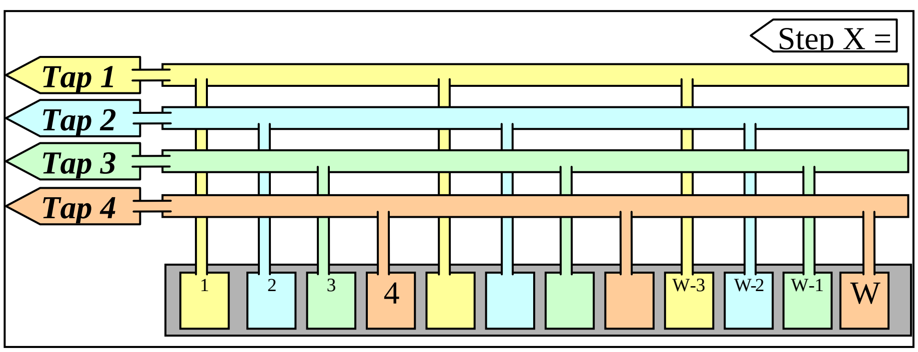


Figure 24-18 Geometry 1X4 (line-scan)

4X-1Y (area-scan)

4 zones in X, 1 zone in Y.

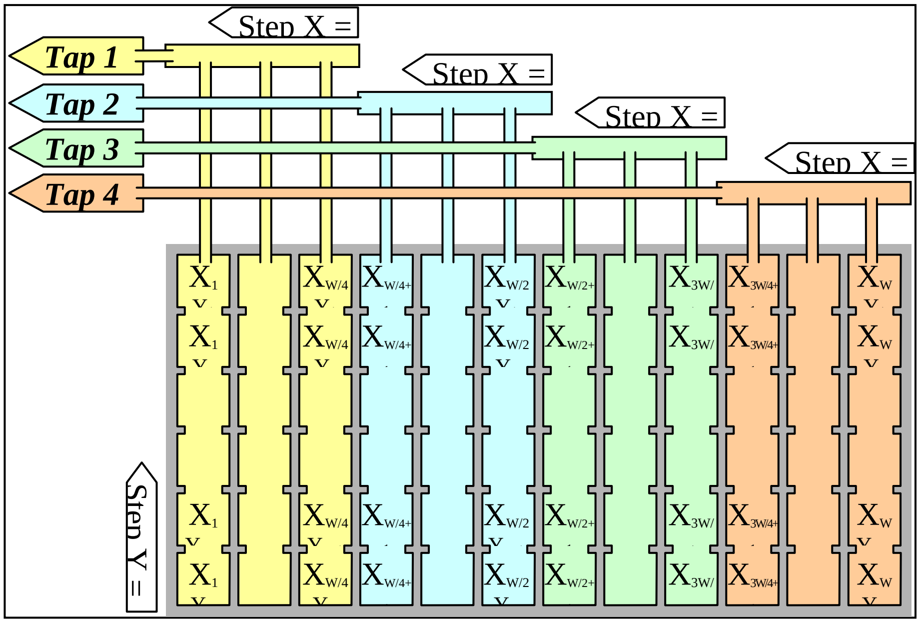


Figure 24-19 Geometry 4X-1Y (area-scan)

4X (line-scan)

4 zones in X.

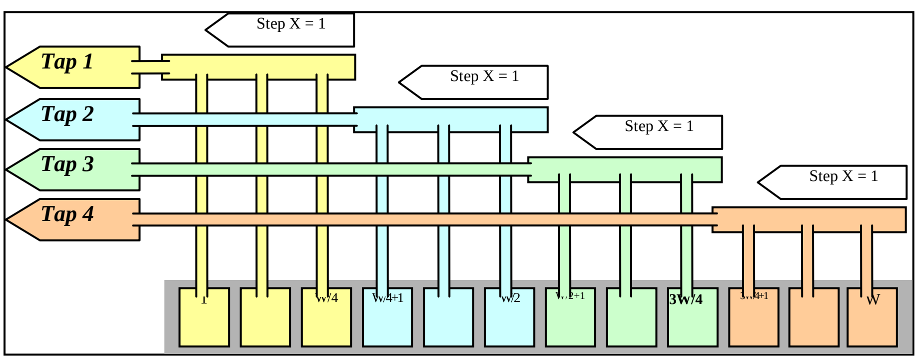


Figure 24-20 Geometry 4X (line-scan)

2X2-1Y (area-scan)

2 zones in X with 2 taps, 1 zone in Y.

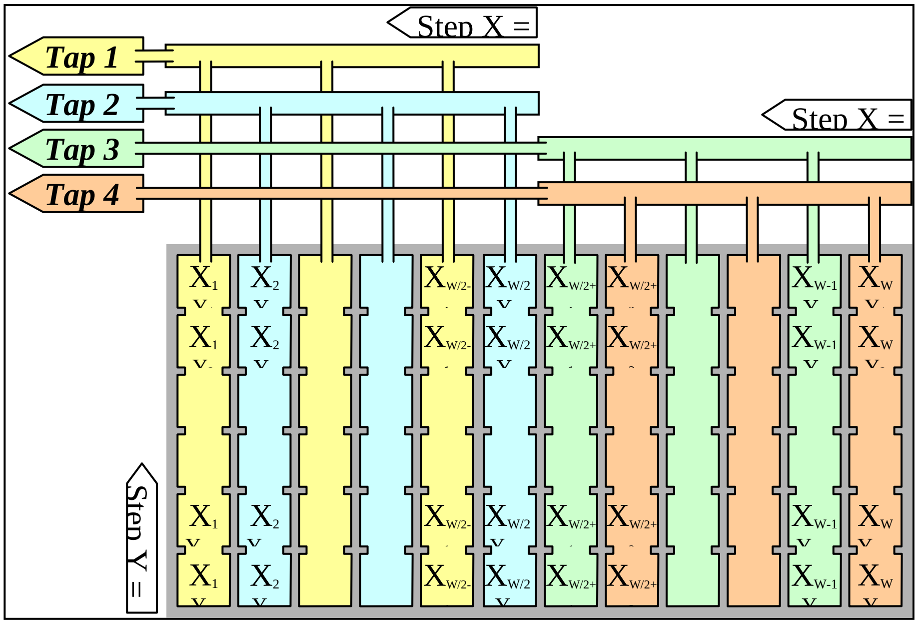


Figure 24-21 Geometry 2X2-1Y (area-scan)

2X2 (line-scan)

2 zones in X with 2 taps.

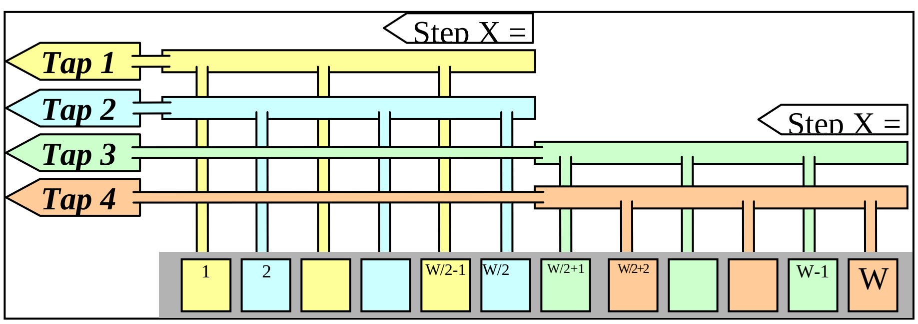


Figure 24-22 Geometry 2X2 (line-scan)

2X2E-1Y (area-scan)

2 zones in X with 2 taps and end extraction, 1 zone in Y.

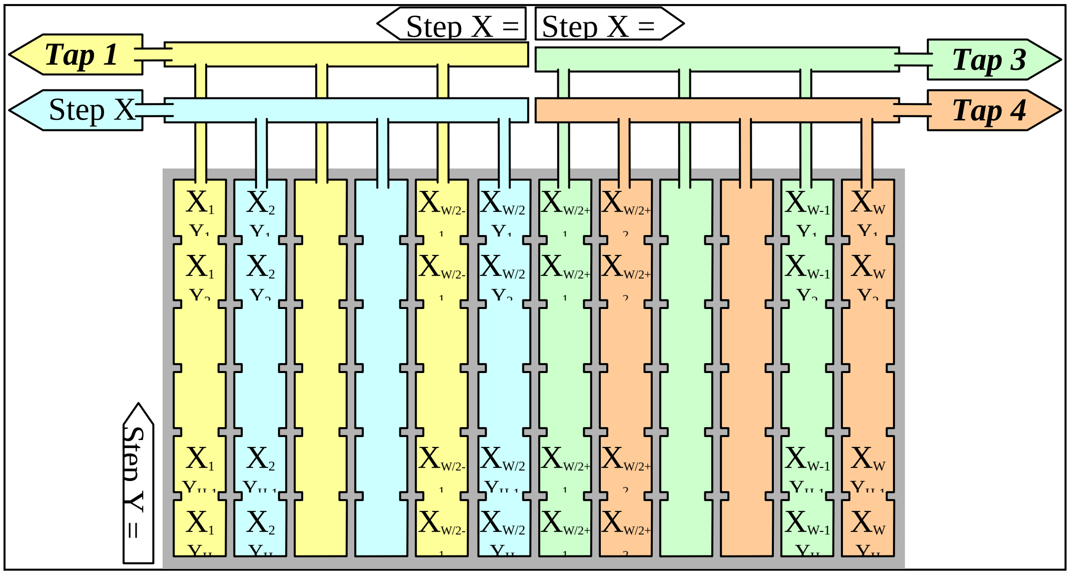


Figure 24-23 Geometry 2X2E-1Y (area-scan)

2X2E (line-scan)

2 zones in X with 2 taps and end extraction.

See similar geometries for building a drawing.

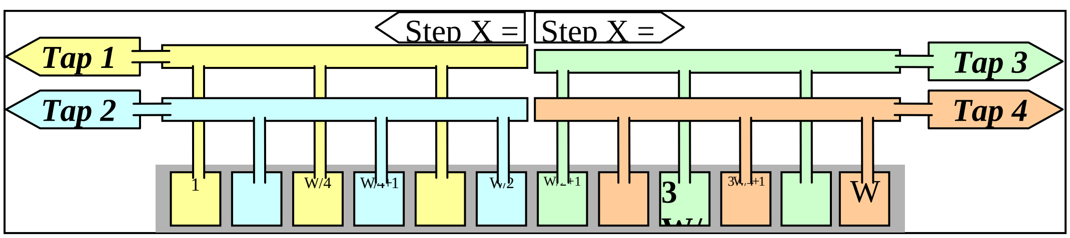


Figure 24-24 Geometry 2X2E (line-scan)

2X2M-1Y (area-scan)

2 zones in X with 2 taps and middle extraction, 1 zone in Y.

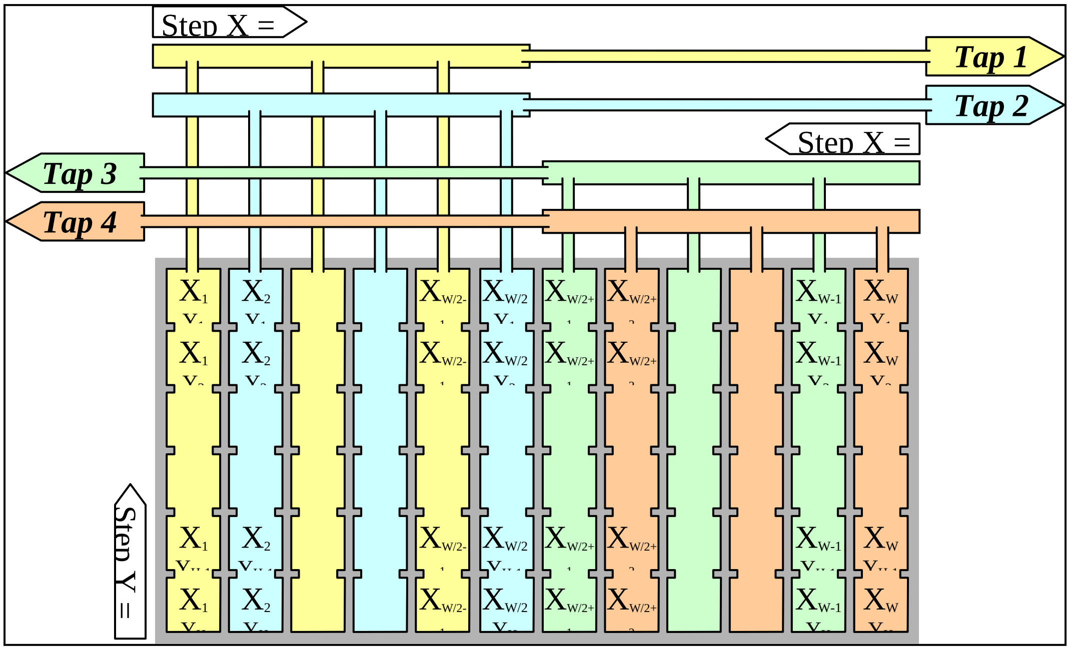


Figure 24-25 Geometry 2X2M-1Y (area-scan)

2X2M (line-scan)

2 zones in X with 2 taps and middle extraction.

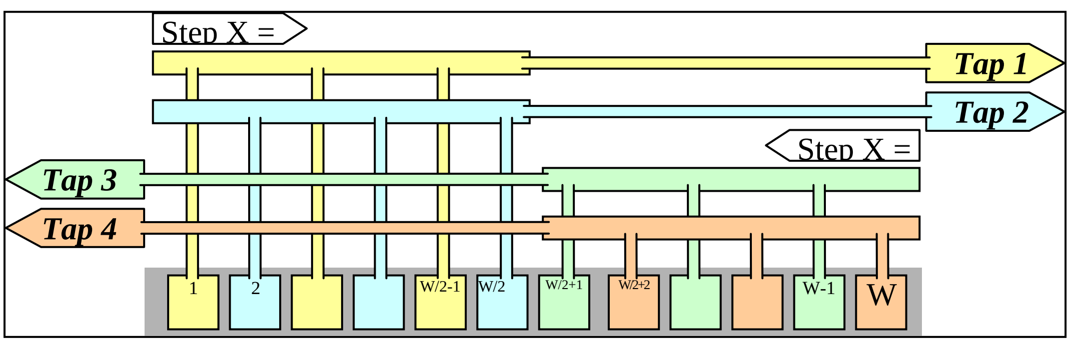


Figure 24-26 Geometry 2X2M (line-scan)

1X2-2YE (area-scan)

1 zone in X with 2 taps, 2 zones in Y with end extraction.

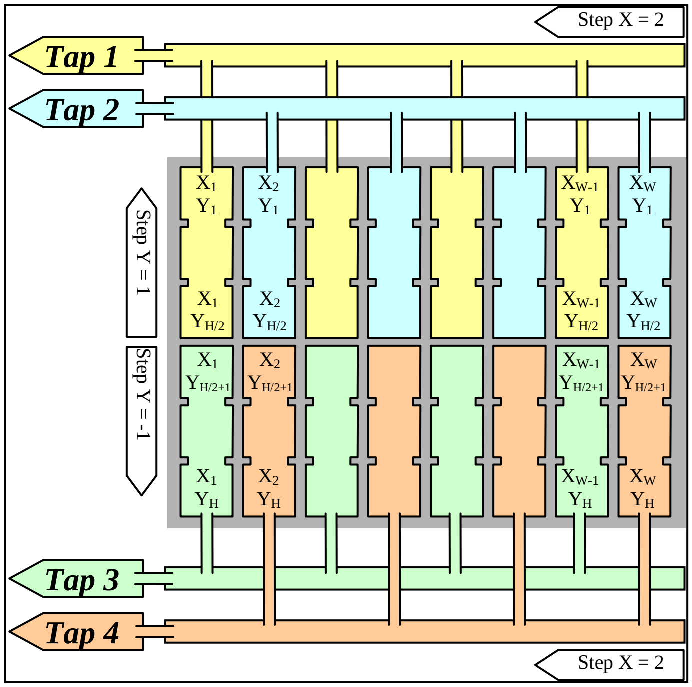


Figure 24-27 Geometry 1X2-2YE (area-scan)

2X-2YE (area-scan)

2 zones in X, 2 zones in Y with end extraction.

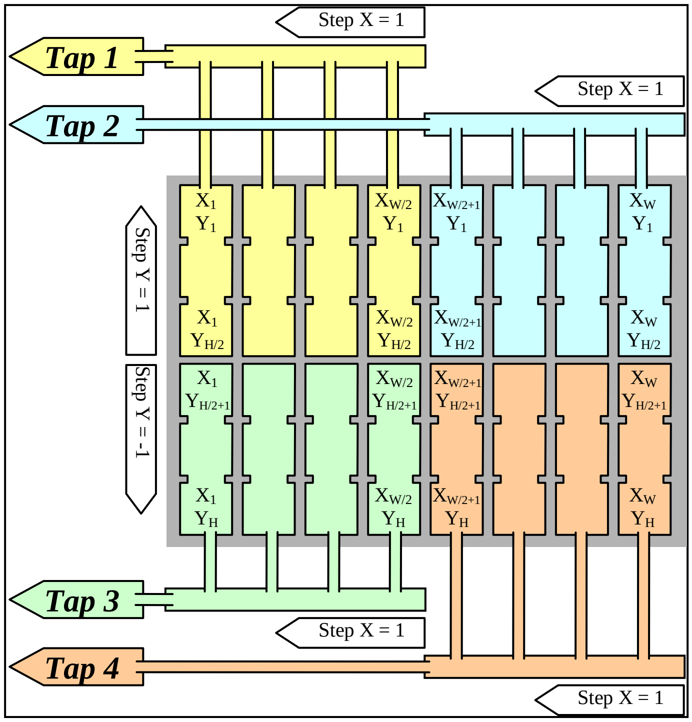


Figure 24-28 Geometry 1X2-2YE (area-scan)

2XE-2YE (area-scan)

2 zones in X with end extraction, 2 zones in Y with end extraction.

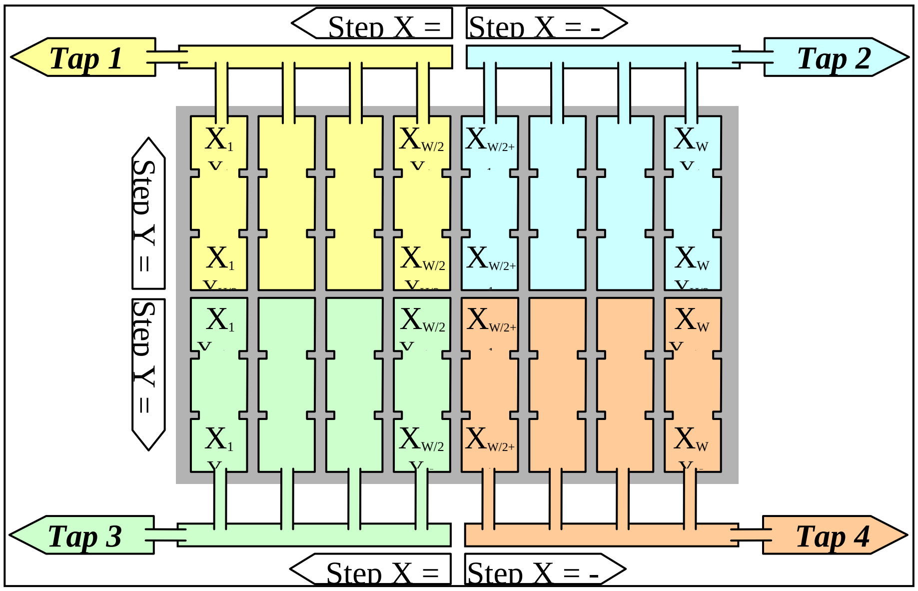


Figure 24-29 Geometry 2XE-2YE (area-scan)

2XM-2YE (area-scan)

2 zones in X with middle extraction, 2 zones in Y with end extraction.

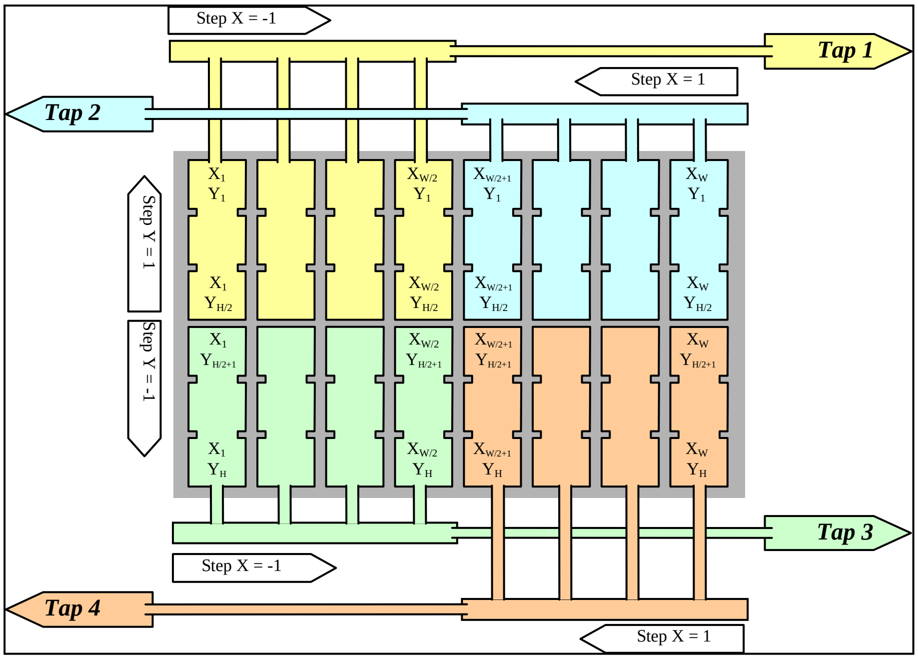


Figure 24-30 Geometry 2XM-2YE (area-scan)

24.6 Octal Tap Geometries

1X8-1Y (area-scan)

1 zone in X with 8 taps, 1 zone in Y.

See similar geometries for building a drawing.

1X8 (line-scan)

1 zone in X with 8 taps.

See similar geometries for building a drawing.

8X-1Y (area-scan)

8 zones in X, 1 zone in Y.

See similar geometries for building a drawing.

8X (line-scan)

8 zones in X.

See similar geometries for building a drawing.

4X2-1Y (area-scan)

4 zones in X with 2 taps, 1 zone in Y.

See similar geometries for building a drawing.

4X2 (line-scan)

4 zones in X with 2 taps.

See similar geometries for building a drawing.

4X2E-1Y (area-scan)

4 zones in X with 2 taps and end extraction, 1 zone in Y.

See similar geometries for building a drawing.

4X2E (line-scan)

4 zones in X with 2 taps and end extraction.

See similar geometries for building a drawing.

2X2E-2YE (area-scan)

2 zones in X with 2 taps, 2 zones in Y with 2 taps.

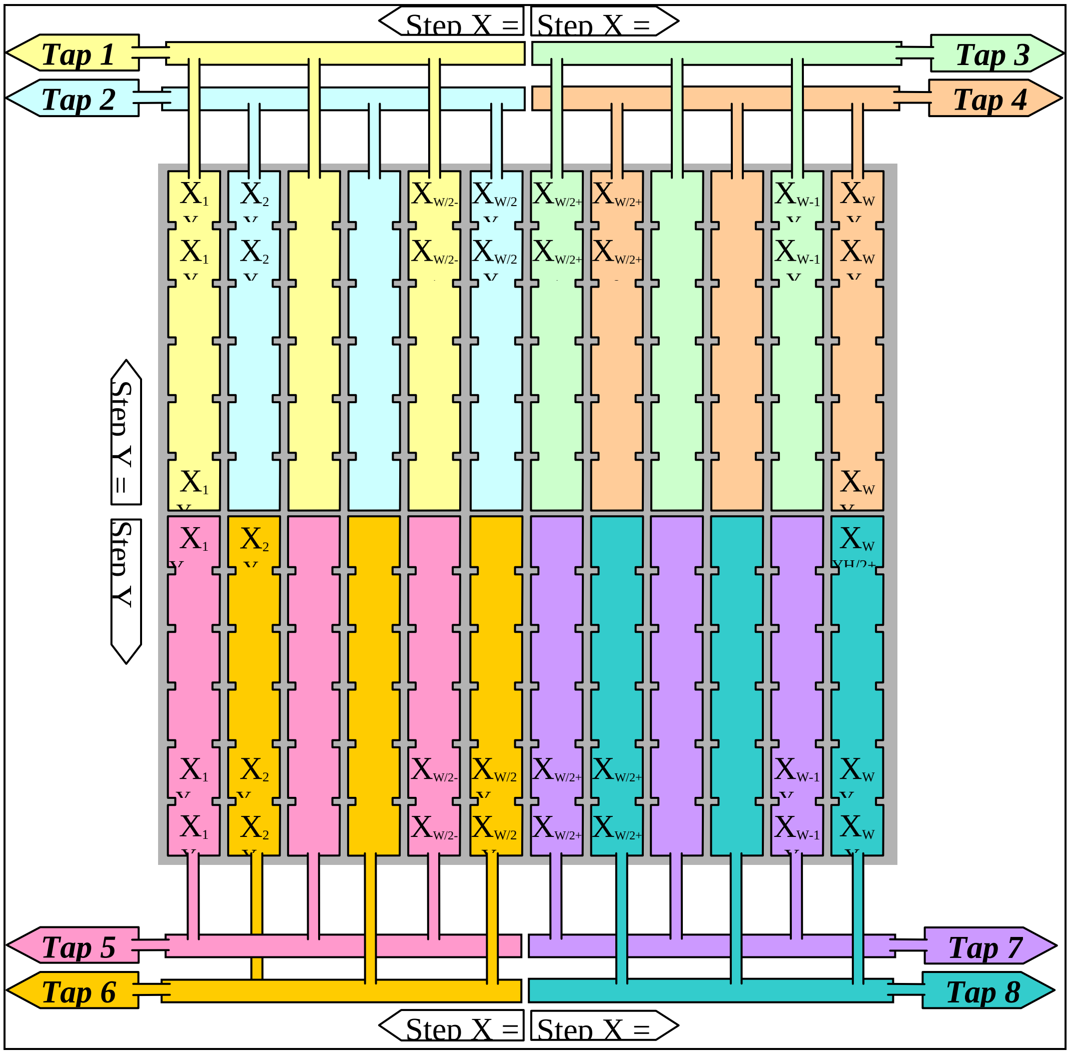


Figure 24-31 Geometry 2X2E-2YE (area-scan)

24.7 Deca Tap Geometries

1X10-1Y (area-scan)

1 zone in X with 10 taps, 1 zone in Y.

See similar geometries for building a drawing.

1X10 (line-scan)

1 zone in X with 10 taps.

See similar geometries for building a drawing.

10X-1Y (area-scan)

10 zones in X, 1 zone in Y.

See similar geometries for building a drawing.

10X (line-scan)

10 zones in X.

See similar geometries for building a drawing.