C4 - Application Note

AutoStart

Rev 1.0

AT - Automation Technology GmbH

Table of Content

Introduction

This application note describes how to use the new AutoStart function of C4-1280-GigE (Firm. Rev. 1.4.1 or higher) and C4-2040-GigE (Firm. Rev. 1.2.2 or higher).

Evaluation Platform

To evaluate each new camera function we strongly recommend using the CX-Explorer due to the best way for configuration and test purposes. Please make sure that you have installed CX-Explorer version 2.0.0 or higher.

Please visit the official online support portal to download the latest version of CX-Explorer.

http://www.automationtechnology.de/sales/

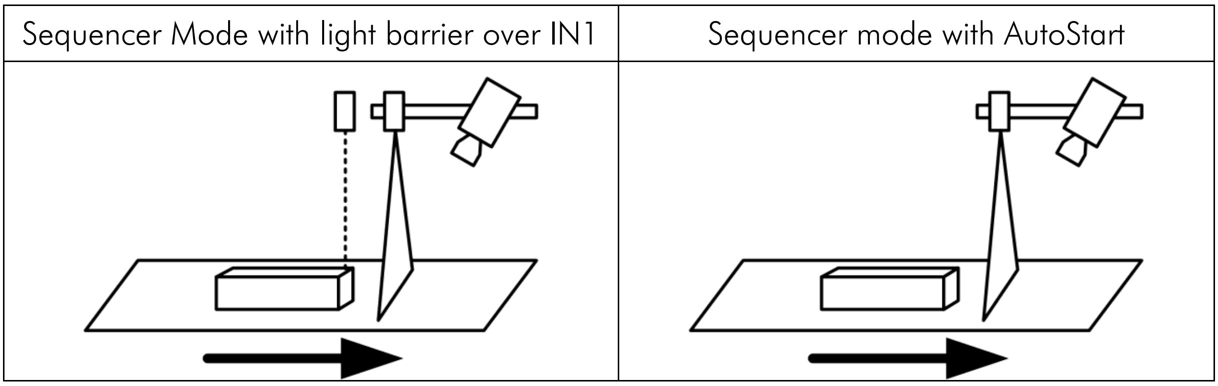
Alternatively please contact the official Automation Technology GmbH Support:

support3d@automationtechnology.de

The AutoStart Mode

This section describes the main concept behind the AutoStart mode and its implementation for optimum usage.

For typical inline inspection systems it is very common to use light barrier for automatic object detection. This has the benefit that the camera will only grab images, when an object is inside the measurement area. However the use of light barriers requires additional hardware and installation as well as higher costs and maintenance. To eliminate this addidional effort the new camera firmware supports the AutoStart mode, which internally simulates a light barrier mechanism without the use of any additional mechanical and electrical components. The figures below demonstrate the difference between a typical inline inspection system with a light barrier and one with the AutoStart function.


Sequencer Mode with light barrier over IN1

Sequencer mode with AutoStart

Figure 1: Triangulation based inline inspection system with light barrier (left) and AutoStart (right)

The AutoStart mode automatically detects the object and triggers the image acquisition to scan the whole object. Furthermore, the AutoStart mode has the option to deliver profile data out of the history buffer, i.e. data which were captured before the initiation of the start trigger and therefore guaranteeing a complete scan. This is very useful when the transition around the geometrical features of the object, which trigger the AutoStart is important.

The AutoStart Condition

This section describes how the AutoStart mode detects the start condition and initiate the start trigger for the profile acquisition.

The idea for the AutoStart mode is that the laser line will move inside a predefined sensor Area Of Interest (AOI) and it will become visible from the camera. The laser line position and the number of valid position values are used as indicators to decide whether a start trigger should be initiated or not. This gives full control over the start trigger. Furthermore, the trigger can be activated at any desired position. The validation parameters of it can be defined by the user.

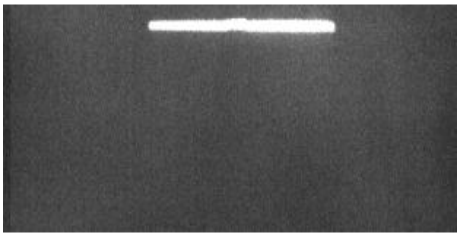
In simple terms the Autostart mode can handle three different situations. In the first case no laser line is visible in the selected AOI, thus the camera is in an idle state and does not transfer any 3D data to the host (PC).

The second case simulates the common “Sequencer Start” mode that is normally performed by a light barrier (e.g. over digital input IN1). Once the laser line is visible inside the AOI and produces valid position values (>0), the camera will capture profiles and transfer these profiles as 3D frames to the host. The host application receives valid camera frames and applies the image processing.

The third case simulates a “smart” light barrier where the user is able to define a specific position inside the AOI at which the start impulse is generated. In this situation the user has full control over the acquisition start and the position values of the first profiles.

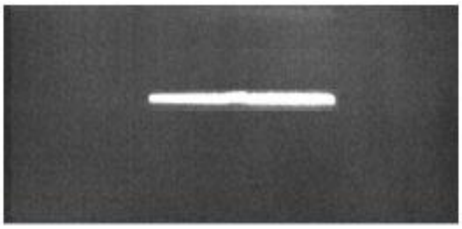
Case I

No laser line is visbile inside the AOI. Consequently no frames are captured and transmitted to the PC.



Case II

Profile acquisition will start as soon as laser line becomes visible inside the AOI. The start trigger is generated immediately.



Case III

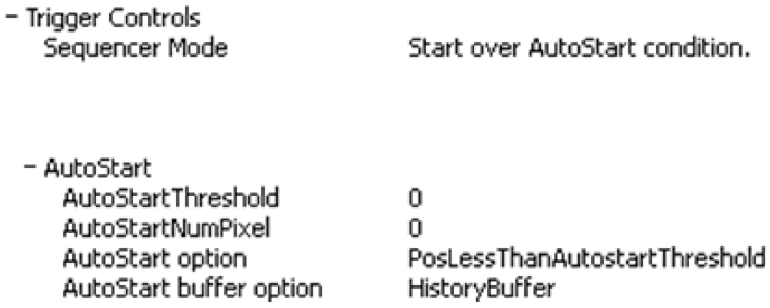
Acquisition will start at a specific position, which is pre-defined by the user. Full control over the start trigger.

The AutoStart Parameters

To fulfil the AuoStart condition additional parameters for the AutoStart mode are included into the GenICam camera description file as shown in the following table:

Name	Type	Visibility	Description
SequencerMode (“AutoStart”)	IEnumeration	Beginner	Symbolic Value “AutoStart”. Start over AutoStart condition.
AutoStart	ICategory	Beginner	Features related to AutoStart
AutoStartThreshold	IBoolean	Beginner	Defines the threshold position within AOI. Valid numbers are for AbsolutePosition = FALSE : 0 to AOI-Height or for AbsolutePosition = TRUE : 0 to Sensor-Height
AutoStartNumPixel	IInteger	Beginner	Minimum number of valid laser positions within AOI required to trigger the Autostart
AutoStartOption	IInteger	Beginner	PosLessThanAutostartThreshold: Start when laser position is less-than AutoStartThreshold PosGreaterThanAutostartThreshold: Start when laser position is greater-than AutoStartThreshold
AutoStartBufferOption	IInteger	Beginner	Defines whether the history buffer should be transmitted as first frame or not: -none -HistoryBuffer

These parameters are also accessible over the XML-Window of the CX-Explorer.



```
- Trigger Controls
   Sequencer Mode                Start over AutoStart condition.


- AutoStart
     AutoStartThreshold          0
     AutoStartNumPixel           0
     AutoStart option            PosLessThanAutostartThreshold
     AutoStart buffer option     HistoryBuffer
```

Figure 2: AutoStart parameters in the XML-window of the CX-Explorer

SequencerMode ("AutoStart")

The AutoStart mode must be activated over the SequencerMode. The symbolic value "AutoStart" will set the AutoStart as the relevant feature to control the sequence.

Without setting the SequencerMode to AutoStart all relevant AutoStart parameters are locked and not accessible over the GenICam access.

AutoStart

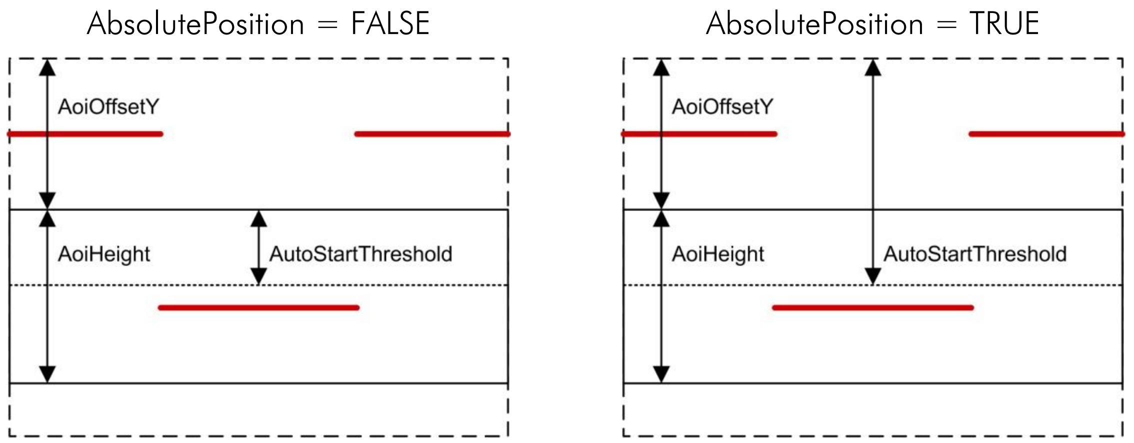
The AutoStart category includes all relevant parameters for the AutoStart mode. The AutoStartThreshold, AutoStartNumPixel and AutoStartOption parameters are required for the start condition. If each condition of these parameters is fulfilled, then the start condition is fulfilled too and the start trigger for the acquisition process is initiated.

```
If(AutoStartThreshold && AutoStartNumPixel && AutoStartOption)
      START
ELSE
      WAIT
END
```

AutoStartThreshold

The AutoStartThreshold parameter defines the laser line position inside the AOI for which the AutoStart condition is fulfilled. For this purpose the laser line position (without sub pixel accuracy) is taken into account. Hence the valid threshold value lies in the range of 0 to AoiHeight.

In case of using the option "AbsolutePosition" (category CameraControls – ModeAndAlgorithmControls - AbsOffsetPos), the threshold value must include the AoiOffsetY value too. The following figure shows the relation between AbsolutePosition and the AutoStartThreshold value.



AbsolutePosition = FALSE

AbsolutePosition = TRUE

Figure 3: Relation between AbsolutePosition and AutoStartThreshold

The following example shows how to set the correct AutoStartThreshold value with respect to the AbsolutePosition in a numerical way.

AoiHeight: 50

AoiOffsetY: 800

Requirement: AutoStartThreshold should be located in the center of the AOI.

AutoStartThreshold = 800 + 50/2 = 825 (Absolute Position = TRUE)

AutoStartThreshold = 50/2 = 25 (Absolute Position = FALSE)

AutoStartNumPixel

In additio to the laser position, which is defined by the AutoStartThreshold, the number of valid laser positions can be taken into account to control the start condition for the AutoStart. This is performed by means of the AutoStartNumPixel parameter. The behavior of this parameter is similar to the known AoiTrackingMinNumPixel node from the AoiTracking mode (see C4 – Application Note for AOI-Search and AOI-Tracking).

As long as the position values of each captured profile do not exceed the number of AutoStartNumPixel, the camera will remain in the idle state and no start trigger will be generated.

AutoStartOption

The AutoStartOption parameter defines the valid range of laser line position. Thus the AutoStartThreshold value defines the border of this range and the AutoStartOption defines whether the range is above or below this threshold value. The following symbolic values are available for the AutoStartOption:

- PosLessThanAutoStartThreshold
- PosGreaterThanAutoStartThreshold

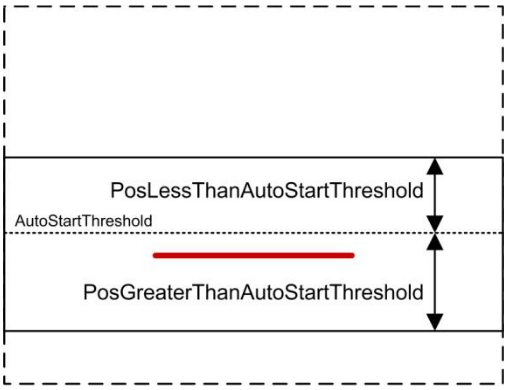

PosLessThanAutoStartThreshold
AutoStartThreshold
PosGreaterThanAutoStartThreshold

Figure 4: AutoStart range

The above figure shows the two different AutoStart ranges, which are defined by the PosLessThanAutostartThreshold and PosGreaterThanAutoStartThreshold.

In case that the AutoStartOption is set to PosLessThanAutoStartThreshold, the start trigger is generated, when the laser line position is **less than** the AutoStartThreshold value.

In case that the AutoStartOption is set to PosGreaterThanAutoStartThreshold, the start trigger is generated, when the laser line position is **greater than** the AutoStartThreshold.

The following examples present a detailed description of the AutoStartOption with respect to the start trigger.

Example#1:

The figure below shows a line plot of a single profile. In this case the laser line position is less than the AutoStartThreshold and the number of pixel that belongs to the laser line is more than the AutoStartNumPixel.

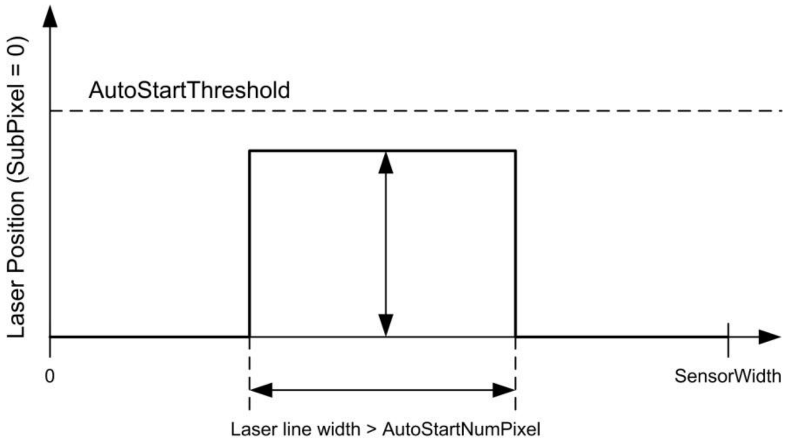

Laser Position (SubPixel = 0)
AutoStartThreshold
0
SensorWidth
Laser line width > AutoStartNumPixel

Figure 5: Line plot of a single profile

This case will cause two different AutoStart behaviours that depend on the AutoStartOption parameter:

IF (AutoStartOption == PosLessThanAutoStartThreshold)

AutoStart status: START impulse is executed, profile acquisition and frame transmission to PC.

IF (AutoStartOption == PosGreaterThanAutoStartThreshold)

AutoStart status: WAIT, internal profile acquisition but **no** frame transmission to PC.

Example#2:

The figure below shows a line plot of a single profile. In this case the laser line position is greater than the AutoStartThreshold and the number of pixel that belongs to the laser line is more than the AutoStartNumPixel.

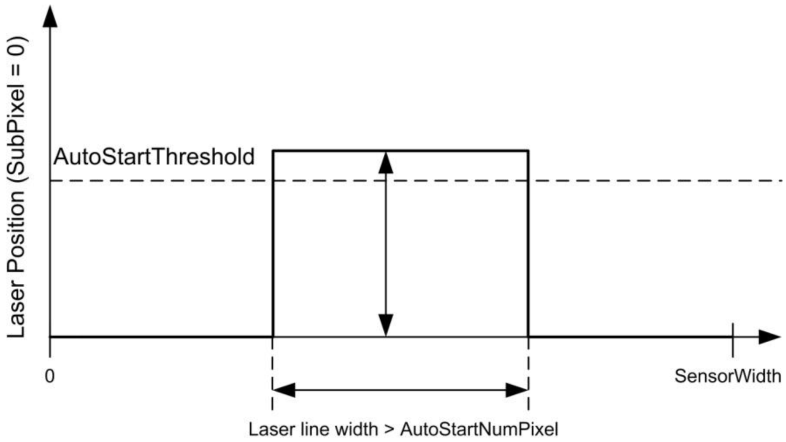

Laser Position (SubPixel = 0)
AutoStartThreshold
0
SensorWidth
Laser line width > AutoStartNumPixel

Figure 6: Line plot of a single profile

This case will cause two different AutoStart behaviours that depend on the AutoStartOption parameter:

IF (AutoStartOption == PosLessThanAutoStartThreshold)

AutoStart status: WAIT, internal profile acquisition but no frame transmission to PC.

IF (AutoStartOption == PosGreaterThanAutoStartThreshold)

AutoStart status: START impulse is executed, profile acquisition and frame transmission to PC.

AutoStartBufferOption

The AutoStartBufferOption parameter has two symbolic values:

- none
- HistoryBuffer

This parameter enables the access of pre-acquired profiles, which were captured before the initiation of start trigger. Further information about this option is available in the main chapter “The AutoStart History Buffer”.

The AutoStart Dependencies and Limitations

The AutoStart mode is a very powerful and smart SequencerMode. However not all special C4 functions are supported in combination with the AutoStart. Therefore the AutoStart does not work in combination with following modes and functions.

- 2D Image Mode (IMG)
- NDR Mode

Furthermore, in case of using multiple AOIs, the relevant AOI for the Autostart evaluation is always the first AOI (AoiSelctor = 1).

The AutoStart History Buffer

The profile that is first acquired after the AutoStart condition is fulfilled is called "trigger profile". The latter is only available in the history buffer, which is transmitted to the PC when the AutoStartBufferOption is set to HistoryBuffer. In addition to the "trigger profile" the history buffer includes also all pre-acquired profiles, i.e. profiles that are internally used for AutoStart evaluation. The number of pre-acquired profiles depends on the frame size and the time period between the first internal profile acquisition and the final "trigger profile". If the profile acquisition before the "trigger profile" is required, then the AutoStartBufferOption parameter must be set to the symbolic value HistoryBuffer. The AutoStartBufferOption has two symbolic values:

- none
- HistoryBuffer

If the profile acquisition only after the start trigger is required, then the history buffer can be disabled by setting the AutoStartBufferOption to "none".

In 3D mode the camera uses three internal buffers to simultaneously capture profiles and transmits 3D frames to the PC. The history buffer is one of these buffers and each buffer has the same size as the total required profiles per frame. The history buffer is used to store the temporal profiles, which are captured during the idle state, i.e. the state in which the camera continuously acquires profiles used for the AutoStart condition evaluation. When one of these profiles fulfils the AutoStart condition, then it is tagged as "trigger profile" and the camera will start with the frame transfer.

The following figure shows the internal use of each camera buffer during the AutoStart beginning. Depending on the AutoStartOption (history buffer on/off) the first frame (Buff#1) is discarded or not.

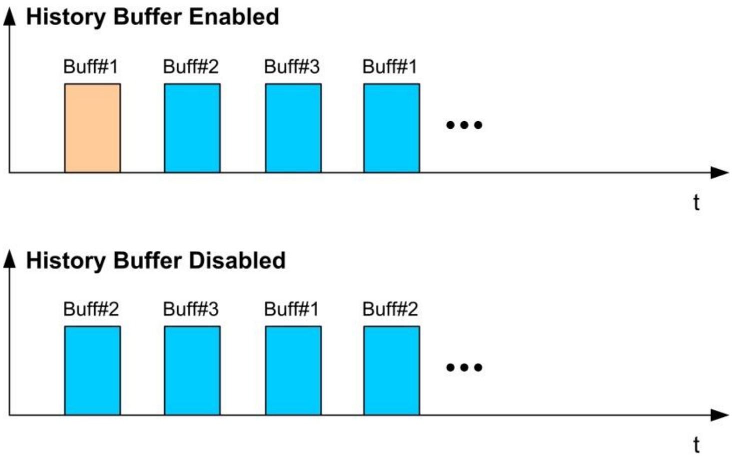


Figure 7: Relationship between internal camera buffer and history buffer

The size of the history buffer is limited. Consequently after a specific number of acquired profiles all free memory is reserved. When the buffer is full, then it starts overwriting the oldest data. In this case a flag in the chunk data is set to indicate whether old data are overwritten or not. Overwriting means that all profiles inside the buffer are valid and represents the intersection between idle state and start.

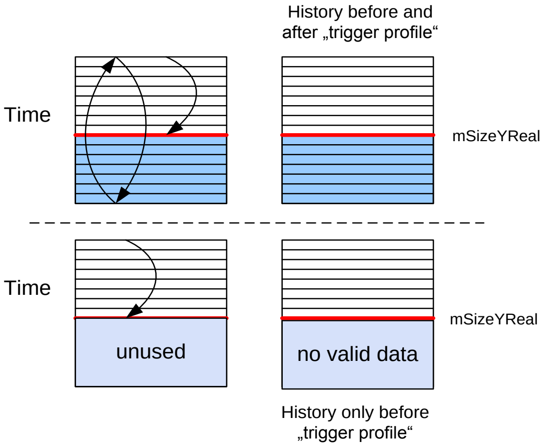

History before and after „trigger profile"
Time
mSizeYReal
Time
unused
no valid data
mSizeYReal
History only before „trigger profile"

Figure 8: Timing and overwriting behavior for the history buffer

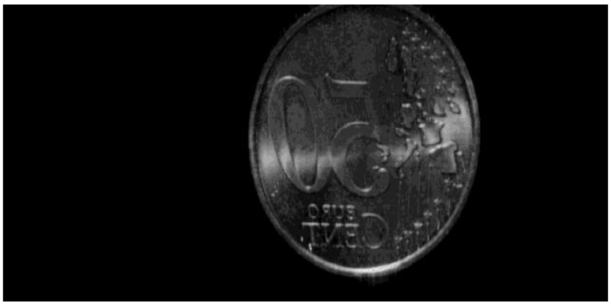
The description of the chunk data is as follows:

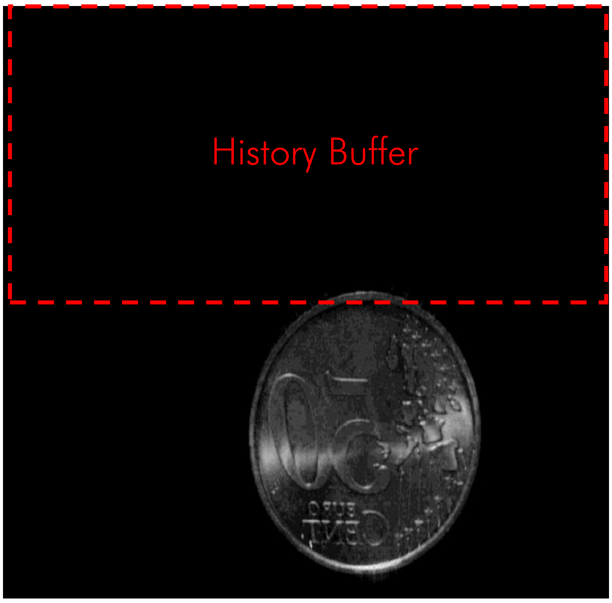
```
typedef struct _GV_ChunkImageInfo
{
unsigned int mSizeYReal;           // number of valid profiles
unsigned int numChunkAcqInfo;
unsigned int flag;                 // flag for indicating overrun status
} GV_ChunkImageInfo;

#define CHUNKIMAGEINFO_FLAG_BIT_BUFFER_OVERRUN       0x04
```

The mSizeYReal value indicates the number of valid profiles from the total 3D frame. In the ideal case the number is equal to the ProfilesPerFrames camera parameter, which means that all data are valid. The flag contains all information with respect to the trigger status and image info. Therefore each bit is reserved for different information. The information about the AutoStart and buffer overrrun condition is defined as CHUNKIMAGEINFO_FLAG_BIT_BUFFER_OVERRUN.

History buffer handling

Without the use of chunk mode of the camera it is not possible to take advantage of the history buffer option. Therefore the chunk mode must be always activated in combination with the usage of the history buffer. The chunk data are required to sort the profiles in the right order. This would be necessary in case of buffer overrun. To simplify the work with the chunk mode and AutoStart mode a C++ example (using Common Vision Blox as image acquisition interface) is provided to demonstrate how to sort the profiles to get a complete 3D scan of an object in the right order. Therefore the tool is able to capture a pre-defined number of frames and concatenate all frames (including the history buffer) to a final 3D frame. This final frame is saved as an image file in TIFF format. The following two images show the result of a 3D scan with and without history buffer option.


AutoStart without History Buffer

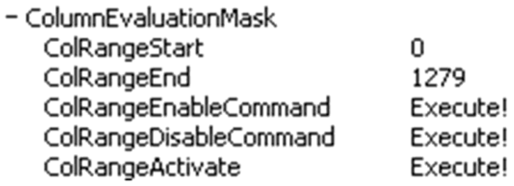

Autostart with (sorted) History Buffer

Column Evaluation Mask

The Column Evaluation Mask is a feature enabling to control the operation of the AutoStart mode. It defines which sensor columns shall be considered as valid for the AutoStart and which should be masked out and excluded from the evaluation.

The use of the Column Evaluation Mask is always required in configurations with AOI width smaller than SensorWidth. Because the AutoStart is performed in the entire sensor width, it is necessary to use the Column Evaluation Mask in order to mask out the sensor columns outside the AOI.

The camera firmware supports two different ways to configure the Column Evaluation Mask. The first one is an XML-based description of different functions, which belongs to the Column Evaluation Mask. Therefore an own category is included into the GenICam camera description file. The big advantage of a standardized feature access via XML is that every 3rd-party software library is able to control these functions due to the fact that they are included and defined as GenICam feature nodes. The next figure shows the graphical representation of these GenICam XML-nodes.



```
- ColumnEvaluationMask
    ColRangeStart              0
    ColRangeEnd                1279
    ColRangeEnableCommand      Execute!
    ColRangeDisableCommand     Execute!
    ColRangeActivate           Execute!
```

Figure 9: ColumnEvaluationMask settings represented as XML-view for GenICam access

The following tree structure defines the location of the “Column Evaluation Mask” features inside the XML-View.

Device Camera → Controls Mode and Algorithm Controls → Column Evaluation Mask

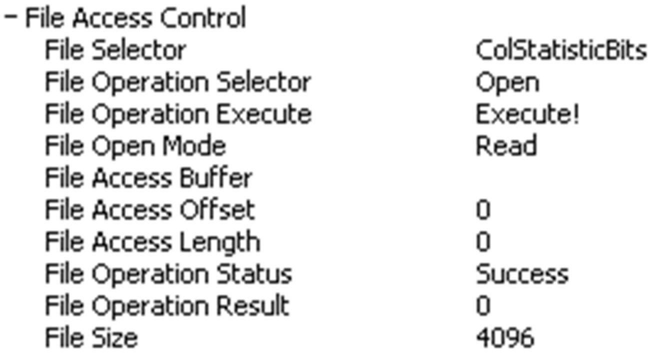
The GenICam name and description of all ColumnEvaluationMask nodes is listed in the table below:

Name	Type	Visibility	Description
ColumnEvaluationMask	ICategory	Beginner	ColumnEvaluation Category. It is a global mask and valid for all functions (AOI-Tracking, AOI-Search, AutoStart)
ColRangeStart	IInteger	Expert	Column start index
ColRangeEnd	IInteger	Expert	Column end index
ColRangeEnableCommand	ICommand	Expert	ENABLE all columns of interval [ColRangeStart, ColRangeEnd]

ColRangeDisableCommand	ICommand	Expert	DISABLE all columns of interval [ColRangeStart, ColRangeEnd]
ColRangeActivate	ICommand	Expert	Activate complete ColumnEvaluationMask and write data into flash memory

Another way to modify the column evaluation mask is to directly write and read over the GenICam File Access Control.

The File Access Control[1] describes all features related to accessing files in the camera. It contains the definition of a generic file access schema for GenICam compliant devices.



```
- File Access Control
    File Selector                 ColStatisticBits
    File Operation Selector       Open
    File Operation Execute        Execute!
    File Open Mode                Read
    File Access Buffer
    File Access Offset            0
    File Access Length            0
    File Operation Status         Success
    File Operation Result         0
    File Size                     4096
```

Figure 10: File Access Control settings represented as XML-view for GenICam access

The following tree structure defines the location of the File Access Control features inside the XML-View.

Device → File Access Control

The file selector "ColStatisticBits" is used to read/write the Column Evaluation Mask in order to reserve the required memory space into the persistent camera memory.

Software solutions, like CommonVisionBlox[2] (CVB) from Stemmer Imaging, provide easy access over their own GenICam File Access Control implementation. For example the following CVB functions can be used to read/write the Column Evaluation Mask from/to the camera:

- NMDownloadFileMemory
  - ➔ Downloads a file from the device which is present in the GenICam::FileSelector and saves it to memory.
- NMUploadFileMemory
  - ➔ Uploads a file to the device which is present in the GenICam::FileSelector and loads it from memory.

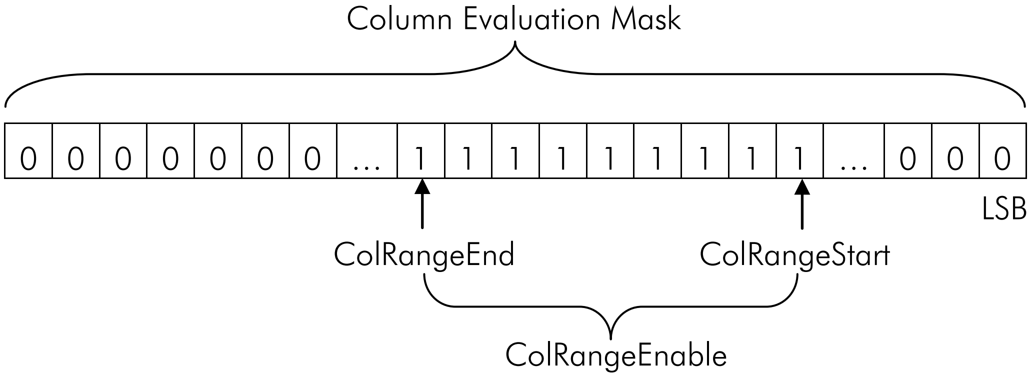
[1] The File Access Control features are summarized in the GenICam Standard Feature Naming Convention (SFNC) http://www.genicam.org

[2] http://www.commonvisionblox.com/

The size of each Column Evaluation Mask depends on the sensor width, which is different between C4-camera models (C4-1280-GigE, C4-2040-GigE, C4-2350-GigE, C4-4090-GigE).

Each Bit represents a single sensor column, which means that the total number of bytes for one column evaluation mask is defined as:

$$size = \frac{SensorWidth}{8} \; Byte$$



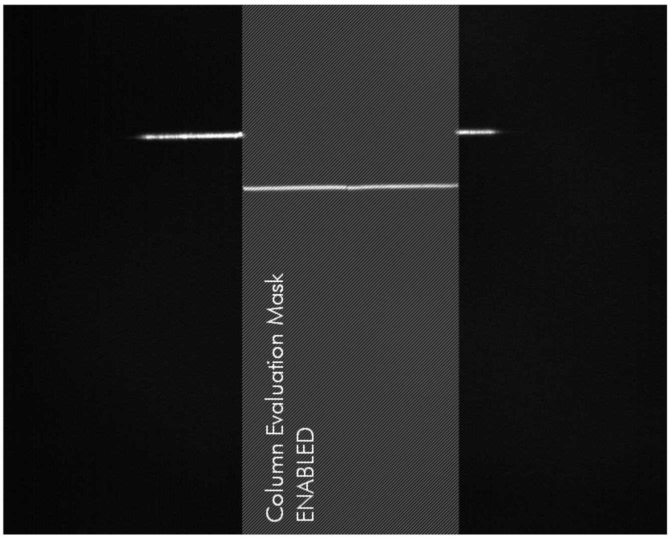


Figure 11: Column Evaluation Mask is enabled for shaded area

Document Revision

Rev. Nr.	Date	Modification
1.0	19.09.2012	First draft
1.1	11.08.2014	Renamed ChunkImageInfo flag to CHUNKIMAGEINFO_FLAG_BIT_BUFFER_OVERRUN