



# Grablink Documentation Update

### Disclaimer

EURESYS s.a. shall retain all property rights, title and interest of the documentation of the hardware and the software, and of the trademarks of EURESYS s.a.

All the names of companies and products mentioned in the documentation may be the trademarks of their respective owners.

The licensing, use, leasing, loaning, translation, reproduction, copying or modification of the hardware or the software, brands or documentation of EURESYS s.a. contained in this book, is not allowed without prior notice.

EURESYS s.a. may modify the product specification or change the information given in this documentation at any time, at its discretion, and without prior notice.

EURESYS s.a. shall not be liable for any loss of or damage to revenues, profits, goodwill, data, information systems or other special, incidental, indirect, consequential or punitive damages of any kind arising in connection with the use of the hardware or the software of EURESYS s.a. or resulting of omissions or errors in this documentation.



# Contents

|  |           |
|--|-----------|
| <b>About This Document</b> .....         | <b>5</b>  |
| Products Scope.....                      | 5         |
| Purpose & Summary.....                   | 5         |
| <br>                                     |           |
| <b>Acquisition</b> .....                 | <b>7</b>  |
| Cluster Mechanism.....                   | 7         |
| <br>                                     |           |
| <b>Pixel Processing Chain</b> .....      | <b>10</b> |
| Bayer CFA to RGB configurations.....     | 10        |
| RGB to RGB configurations.....           | 12        |
| White Balance Operator.....              | 14        |
| <br>                                     |           |
| <b>White Balance</b> .....               | <b>16</b> |
| What Is White Balance?.....              | 16        |
| Automatic White Balance.....             | 17        |
| Automatic Calibration Description.....   | 17        |
| Automatic Calibration Requirements.....  | 17        |
| Automatic Calibration Timing.....        | 18        |
| AWB_AREA Settings Description.....       | 19        |
| <br>                                     |           |
| <b>Image Geometrical Operators</b> ..... | <b>20</b> |
| Image Cropping.....                      | 20        |
| Image Flipping Operators.....            | 22        |
| <br>                                     |           |
| <b>Metadata Insertion</b> .....          | <b>23</b> |
| Short Description.....                   | 23        |
| Functional Description.....              | 24        |
| Metadata Insertion.....                  | 24        |
| Sampling Time.....                       | 24        |

|                                   |    |
|-----------------------------------|----|
| System I/O Input State.....       | 24 |
| Motion Encoder Pulse Counter..... | 24 |
| LVAL Pulse Counter.....           | 25 |

**MultiCam Parameters Reference Update..... 26**

|  |    |
|--|----|
| Channel Class - Grabber Configuration Category.....  | 26 |
| MetadataInsertion.....                               | 26 |
| Channel Class - Grabber Timing Category.....         | 28 |
| GrabWindow.....                                      | 28 |
| WindowX_Px.....                                      | 29 |
| WindowY_Ln.....                                      | 31 |
| OffsetX_Px.....                                      | 32 |
| OffsetY_Ln.....                                      | 34 |
| WindowOrgX_Px.....                                   | 35 |
| WindowOrgY_Ln.....                                   | 37 |
| Channel Class - White Balance Operator Category..... | 38 |
| WBO_Mode.....  | 38 |
| WBO_GainR.....                                       | 40 |
| WBO_GainG.....                                       | 41 |
| WBO_GainB.....                                       | 42 |
| WBO_Width.....                                       | 44 |
| WBO_Height.....                                      | 45 |
| WBO_OrgX.....  | 47 |
| WBO_OrgY.....  | 48 |
| WBO_Status.....                                      | 50 |
| Channel Class - Cluster Category.....                | 51 |
| MaxFillingSurfaces.....                              | 51 |
| ImageFlipX.....                                      | 53 |
| ImageFlipY.....                                      | 54 |
| Surface Class - Surface Dynamics Category.....       | 55 |
| SurfaceState.....                                    | 55 |

# About This Document

## Products Scope

---

This document applies to the following products:

| Product Code | Product Designation |
|--------------|---------------------|
| 1624         | Grablink Base       |
| 1623         | Grablink DualBase   |
| 1622         | Grablink Full       |
| 1626         | Grablink Full XR    |

## Purpose & Summary

---

This document is an update of the MultiCam Boards Documentation that was provided with MultiCam 6.7.

It provides public information about the following product updates:

### Improved Grablink Base, DualBase, Full and Full XR driver robustness vs. latencies

On Grablink Base, DualBase, Full and Full XR boards, MultiCam introduces the new MaxFillingSurfaces Cluster parameter to allow a maximum number of surfaces to be put in the FILLING state simultaneously. When doing so, the robustness of the acquisition and signalling mechanisms in case of system latencies is increased. By default, MaxFillingSurfaces=MAXIMUM and this mechanism is enabled. The previous operation mode where only one surface can be in the FILLING state at a time can be restored by setting MaxFillingSurfaces to MINIMUM.

*This improvement is available since MultiCam 6.8.2.2065.*

### White Balance Operator available on Grablink Base, Grablink DualBase, Grablink Full and Grablink Full XR

A White Balance Operator providing automatic (WBO\_Mode=ONCE) and manual (WBO\_Mode=MANUAL) calibration methods is now available for Bayer CFA and RGB color cameras on Grablink Base, Grablink DualBase, Grablink Full and Grablink Full XR boards.

*This new feature is available since MultiCam 6.8.1.1986.*

### Metadata Insertion available on Grablink Full XR

The first 10 bytes of each image line are replaced by a fixed set of metadata including:

- ❖ The logical state of all (6) System I/O input lines
- ❖ The value of the motion encoder pulse counter
- ❖ The value of the Camera Link LVAL pulse counter

*This new feature is available since MultiCam 6.9.1.2271.*

# Acquisition

## Cluster Mechanism

---

### Surface states

To implement the cluster mechanism, MultiCam uses the state of the surface, which is available through the MultiCam parameter *SurfaceState*.

Any instantiated surface is necessarily in one of the five following states:

| Surface State | Description   |
|---------------|---|
| FREE          | The surface is unconditionally able to receive image data from the grabber.                       |
| FILLING       | The surface is presently receiving or ready to receive image data from the grabber.               |
| FILLED        | The surface has finished receiving image data from the grabber, and thus is ready for processing. |
| PROCESSING    | The surface is being processed by the host processor.   |
| RESERVED      | The surface is removed from the standard state transition mechanism.                              |

The state of the surface is unique in the sense that, at a given instant, a surface belonging to several clusters is perceived in a consistent state by all the associated channels.

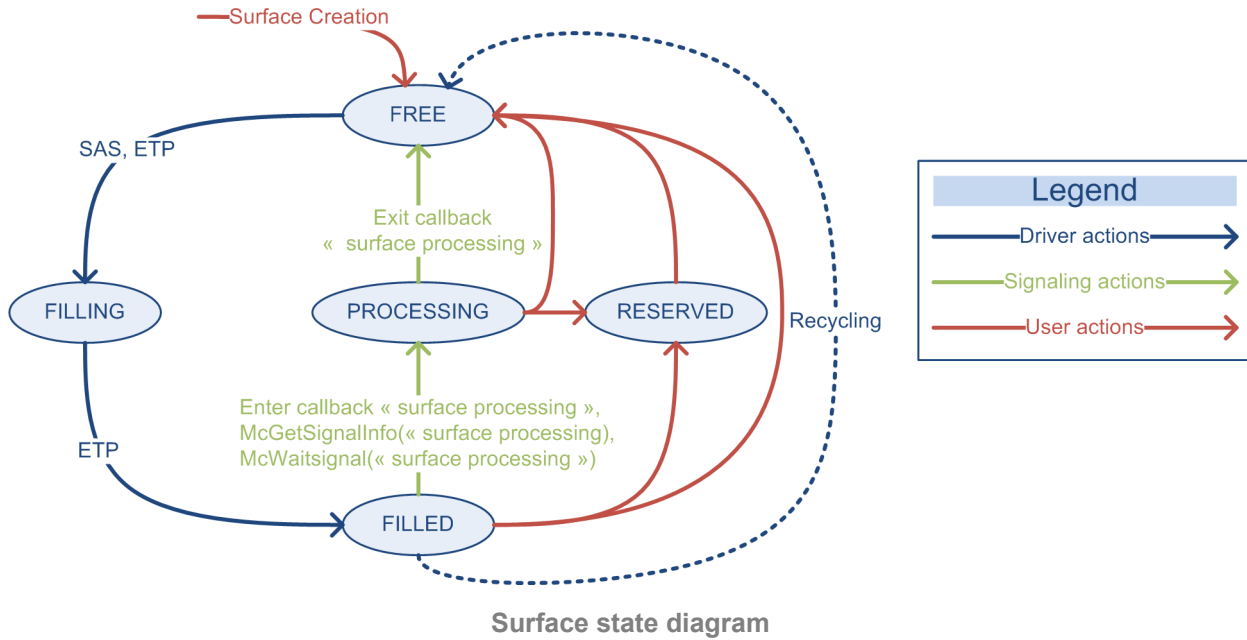
## Surface state transitions

State transitions table

| State origin         | State destination | Initiator                            | Occurrence and applicability  |
|----------------------|-------------------|--------------------------------------|---|
| Not applicable       | FREE              | User application                     | On creation of a new surface.   |
| FREE                 | FILLING           | MultiCam driver                      | On Start Acquisition Sequence and End of Transfer Phase events: <ul style="list-style-type: none"> <li>❖ When <code>MaxFillingSurfaces = MAXIMUM</code>: applies to all (up to 512) FREE surfaces in the cluster</li> <li>❖ Otherwise applies to only one FREE surface</li> </ul>   |
| FILLING              | FILLED            | MultiCam driver                      | On End of Transfer Phase event: <ul style="list-style-type: none"> <li>❖ When <code>MaxFillingSurfaces = MAXIMUM</code>: applies to all (up to 512) FILLING surfaces in the cluster that have finished receiving image data from the grabber, and thus are ready for processing</li> <li>❖ Otherwise applies to the unique FILLING surface</li> </ul>                                   |
| FILLED               | PROCESSING        | User application or Operating System | Applies to the oldest FILLED surface (if any) when the cluster contains no more PROCESSING surface and <ul style="list-style-type: none"> <li>❖ on Execution of the <code>McGetSignalInfo</code> function or</li> <li>❖ on release (exit) of the <code>McWaitSignal(SurfaceProcessing)</code> function or</li> <li>❖ on entry of the "Surface Processing" callback function.</li> </ul> |
| PROCESSING           | FREE              | User application or Operating System | Applies to the unique PROCESSING surface (if any) when: <ul style="list-style-type: none"> <li>❖ setting the parameter <code>MC_SurfaceState</code> to the value <code>MC_SurfaceState_FREE</code> or</li> <li>❖ automatically when exiting the "Surface Processing" callback function.</li> </ul>  |
| FILLED<br>RESERVED   | FREE              | User application                     | Applies to any FILLED or RESERVED surface when setting the parameter <code>MC_SurfaceState</code> to the value <code>MC_SurfaceState_FREE</code> .  |
| FILLED<br>PROCESSING | RESERVED          | User application                     | Applies to any FILLED or PROCESSING surface when setting the parameter <code>MC_SurfaceState</code> to the value <code>MC_SurfaceState_RESERVED</code> .  |
| FILLED               | FREE              | MultiCam driver                      | Applies to the oldest FILLED surface (if any) when the cluster contains no more FREE surfaces   |

The following drawing shows a simplified state transition diagram applying to any surface in the cluster:





For a cluster having a total of N registered surfaces:

- ❖ 0 up to N surfaces can be in the FREE state
- ❖ 0 up to N (limited to 512) surfaces can be in the FILLING state when the parameter `MaxFillingSurfaces = MAXIMUM`
- ❖ 0 or 1 surface can be in the FILLING state when the parameter `MaxFillingSurfaces = MINIMUM`
- ❖ 0 up to N surfaces can be in the FILLED state
- ❖ 0 or 1 surface can be in the PROCESSING state
- ❖ 0 to (N-2) surfaces can be in the RESERVED state

**Note.** There is at most one surface in the PROCESSING state per cluster!

**Note.** At least 2 surfaces should be left outside the RESERVED state to maintain a minimal operability of the cluster mechanism.

#### Related Links

**MultiCam reference topic(s)**

[MaxFillingSurfaces](#) on page 51

[SurfaceState](#) on page 55

# Pixel Processing Chain

The acquisition channels of Grablink Base, Grablink DualBase, Grablink Full, and Grablink Full XR implement a pixel processing chain. It is made of the following processing elements:

- ❖ A Bayer CFA Decoder
- ❖ A White Balance Operator
- ❖ A Look-Up table Operator

The pixel processing chain is configurable; each element can be individually configured, enabled or disabled. However the order of the elements cannot be modified!

## Related Links

[Bayer CFA to RGB configurations](#) on page 10

[RGB to RGB configurations](#) on page 12

[White Balance Operator](#) on page 14

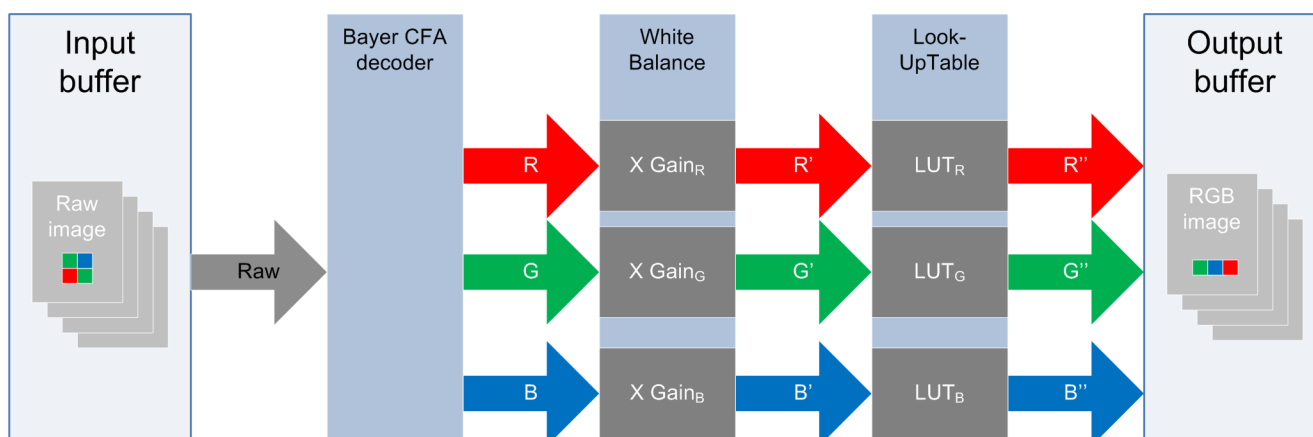
## Bayer CFA to RGB configurations

In the Bayer CFA to RGB configurations, the pixel processing chain uses the following elements:

- ❖ Bayer CFA Decoder
- ❖ White Balance operator (Optional)
- ❖ Look-Up-table operator (Optional)

The Look-Up-table operator is configured for RGB color processing.

The processing chain outputs one RGB pixel for each RAW pixel of the input buffer.



Pixel Processing Chain - Bayer CFA => RGB Configuration

**Bayer CFA to RGB Configurations - Availability and Performance**

| Pixel Input Format | WBO      | LUT                  | Pixel Output Format(s)                       | Compatible products and peak pixel rate (Megapixels/sec)  |
|--------------------|----------|----------------------|--|---|
| Bayer8             | Optional | 8-bit to 8-bit RGB   | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Base:</b> 125<br><b>Grablink DualBase:</b> 125<br><b>Grablink Full:</b> 250<br><b>Grablink Full XR:</b> 250 |
| Bayer10            | Optional | 10-bit to 8-bit RGB  | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Base:</b> 125<br><b>Grablink DualBase:</b> 125<br><b>Grablink Full:</b> 250<br><b>Grablink Full XR:</b> 250 |
| Bayer10            | Optional | 10-bit to 10-bit RGB | RGB30 Planar                                 | <b>Grablink Full:</b> 125<br><b>Grablink Full XR:</b> 125   |
| Bayer10            | Optional | 10-bit to 16-bit RGB | RGB48 Planar                                 | <b>Grablink Full:</b> 125<br><b>Grablink Full XR:</b> 125   |
| Bayer12            | Optional | 12-bit to 8-bit RGB  | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Base:</b> 125<br><b>Grablink DualBase:</b> 125<br><b>Grablink Full:</b> 250<br><b>Grablink Full XR:</b> 250 |
| Bayer12            | Optional | 12-bit to 12-bit RGB | RGB36 Planar                                 | <b>Grablink Full:</b> 125<br><b>Grablink Full XR:</b> 125   |
| Bayer12            | Optional | 12-bit to 16-bit RGB | RGB48 Planar                                 | <b>Grablink Full:</b> 125<br><b>Grablink Full XR:</b> 125   |
| Bayer14            | Optional | 12-bit to 8-bit RGB  | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Base:</b> 125<br><b>Grablink DualBase:</b> 125<br><b>Grablink Full:</b> 250<br><b>Grablink Full XR:</b> 250 |
| Bayer14            | Optional | 12-bit to 14-bit RGB | RGB42 Planar                                 | <b>Grablink Full:</b> 125<br><b>Grablink Full XR:</b> 125   |
| Bayer14            | Optional | 12-bit to 16-bit RGB | RGB48 Planar                                 | <b>Grablink Full:</b> 125<br><b>Grablink Full XR:</b> 125   |
| Bayer16            | Optional | 12-bit to 8-bit RGB  | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Base:</b> 125<br><b>Grablink DualBase:</b> 125<br><b>Grablink Full:</b> 250<br><b>Grablink Full XR:</b> 250 |
| Bayer16            | Optional | 12-bit to 16-bit RGB | RGB48 Planar                                 | <b>Grablink Full:</b> 125<br><b>Grablink Full XR:</b> 125   |

## RGB to RGB configurations

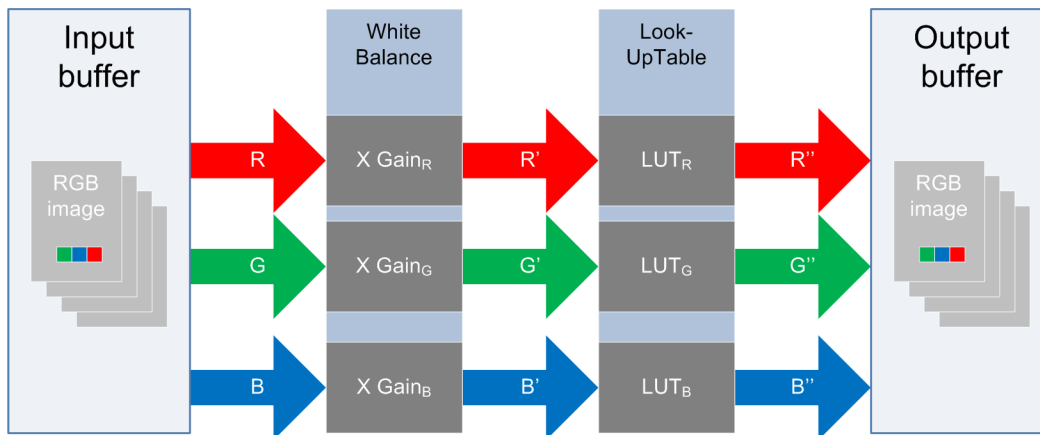
---

In the RGB to RGB configurations, the pixel processing chain uses the following elements:

- ❖ White Balance operator (Optional)
- ❖ Look-Up-table operator (Optional)

The Look-Up-table Operator is configured for RGB color processing.

The processing chain outputs one RGB pixel for each RGB pixel of the input buffer.



Pixel Processing Chain - RGB => RGB Configuration

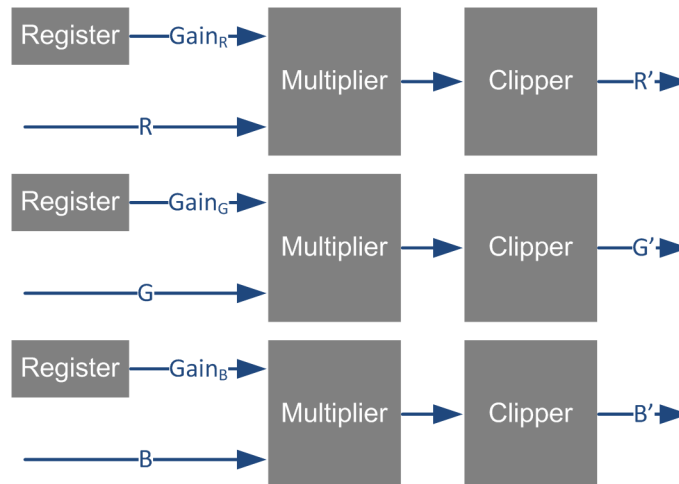
**RGB to RGB Configurations - Availability and Performance**

| Pixel Input Format | WBO      | LUT                  | Pixel Output Format(s)                       | Compatible products and peak pixel rate (Megapixels/sec)  |
|--------------------|----------|----------------------|--|---|
| RGB24              | Optional | 8-bit to 8-bit RGB   | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Base:</b> 125(85*)<br><b>Grablink DualBase:</b> 125(85*)<br><b>Grablink Full:</b> 250(170*)<br><b>Grablink Full XR:</b> 250(170*) |
| RGB30              | Optional | 10-bit to 8-bit RGB  | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Full:</b> 250(85*)<br><b>Grablink Full XR:</b> 250(85*)   |
| RGB30              | Optional | 10-bit to 10-bit RGB | RGB30 Planar                                 | <b>Grablink Full:</b> 125(85*)<br><b>Grablink Full XR:</b> 125(85*)   |
| RGB30              | Optional | 10-bit to 16-bit RGB | RGB48 Planar                                 | <b>Grablink Full:</b> 125(85*)<br><b>Grablink Full XR:</b> 125(85*)   |
| RGB36              | Optional | 12-bit to 8-bit RGB  | RGB24 Packed<br>RGB32 Packed<br>RGB24 Planar | <b>Grablink Full:</b> 250(85*)<br><b>Grablink Full XR:</b> 250(85*)   |
| RGB36              | Optional | 12-bit to 10-bit RGB | RGB36 Planar                                 | <b>Grablink Full:</b> 125(85*)<br><b>Grablink Full XR:</b> 125(85*)   |
| RGB36              | Optional | 12-bit to 16-bit RGB | RGB48 Planar                                 | <b>Grablink Full:</b> 125(85*)<br><b>Grablink Full XR:</b> 125(85*)   |

**Note.** (xx\*) indicates the highest pixel rate achievable by Camera Link for the targeted board.

## White Balance Operator

---



White Balance Operator - Block Diagram

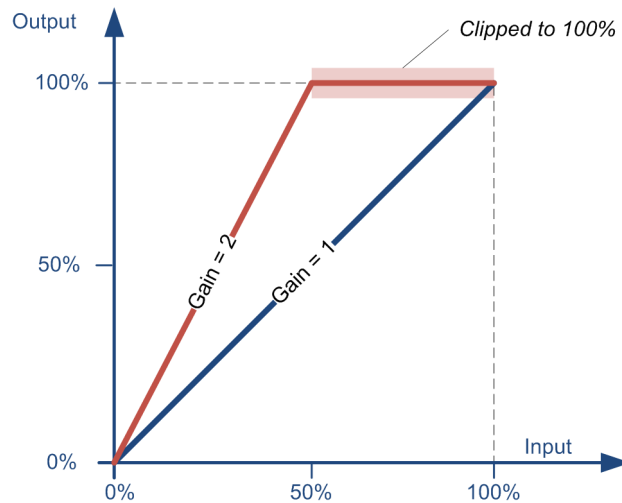
The White Balance Operator is an element of the pixel processing chain. It is composed with 3 identical processing blocks, one for each color component. Each processing block contains 3 elements:

- ❖ One register
- ❖ One multiplier
- ❖ One clipper

The register element holds the gain correction factor. The gain value is registered as a 16-bit unsigned binary value allowing gain correction factors to be accurately defined.

The multiplier computes the product of the gain correction factor and the color component value. It is capable to handle components having 8-bit, 10-bit, 12-bit, 14-bit and 16-bit bit depth.

The multiplier output is clipped to the maximum value of the digital output scale. The digital output scale is in all cases identical to the digital input scale; itself identical to the digital output scale of the camera. For instance, for a camera delivering 10-bit components, the digital scale is [0..1023].



**White Balance Operator - Transfer Function**

The above drawing shows 2 transfer functions of one component of the White Balance Operator:

- ❖ The blue line corresponds to a gain setting of 1.000; i.e., the minimal allowed gain value.
- ❖ The red line corresponds to a gain setting of 2.000. The output remains proportional to the input until the 100% full-scale output is reached; for greater input values, the output is clipped to 100% full-scale!

#### **Related Links**

##### **MultiCam reference topic(s)**

[WBO\\_Mode](#) on page 38

[WBO\\_GainR](#) on page 40

[WBO\\_GainG](#) on page 41

[WBO\\_GainB](#) on page 42

# White Balance

## What Is White Balance?

---

### Color image acquisition

A color image acquisition involves the use of three color filters on the camera sensor. Each color filter restricts the light source to a range of wavelengths of the light spectrum, either red (R), green (G), or blue (B).

An ideal capture system renders a white object as a white image. A white stimulation should yield the same signal for R, G and B filters. But practically, there are always unavoidable defects on the signals that introduce a **white imbalance**.

### White imbalance factor

Several factors, due to the camera and to the capture conditions, are responsible for the white imbalance:

- ❖ Object illumination. The color of an object is a combination of its reflectivity and the spectral contents of the illuminating light.
- ❖ Camera optical filters response.
- ❖ Sensor sensitivity, which is not the same for the three ranges of wavelength.
- ❖ Different gain coefficients applied to each color signal before digitization.

### White balance correction

MultiCam can correct the white imbalance of the capture system. The operation is called the **white balance**:

- ❖ The **white balance operator** applies correcting coefficients (R, G, and B gains) to each color signal, so, for a white object, the combination of the R, G, and B signals renders a white image.
- ❖ The **white balance calibration** is the computation of the three R, G, and B gains. It is performed on a representative image area, prior to the image capture. It can be automatic or manual.

### Related Links

#### MultiCam reference topic(s)

[WBO\\_Mode](#) on page 38

[White Balance Operator](#) on page 14

[Automatic Calibration Description](#) on page 17





# Automatic White Balance

---

## Automatic Calibration Description

The color calibration process takes place during the first acquisition phase of a MultiCam acquisition sequence when the `WBO_Mode` is set to `ONCE`.

The color calibrator analyzes a rectangular area (`AWB_AREA`) of one uncorrected image and computes a correcting gain factor for each RGB color component.

The correction factor for the color component having the strongest response is always 1; the correction factors for the weakest color components are greater than 1.

Providing that the requirements of the **color source equipment**, the **calibration target** and the **acquisition channel settings** are fulfilled, the calibrator estimates the gain factors with an accuracy better than 1/1000.

Applying the calculated gain correction factors to the White Balance Operator for subsequent image acquisitions allows on-the-fly color balancing of the acquired images.

The calibrator returns a `NOT_OK` status in the following cases:

- ❖ Excessive color imbalance.
- ❖ Not enough pixels satisfying the calibration target requirements in the `AWB_AREA`.

## Automatic Calibration Requirements

This topic describes the requirements that must be fulfilled to obtain optimal calibration results.

### Image Source Equipment Requirements

The image source equipment including: the camera, the lighting and the optical elements, must exhibit:

- ❖ A linear response: The digital value of each color component must be proportional to the light intensity of the corresponding color.
- ❖ A moderate color imbalance: The ratio between the response of the strongest color component and the weakest color component must be less than 5.

### Calibration Target Requirements

The calibration target is a neutral color object located in the field of view of the camera during the calibration process.

The form of the target can be either:

- ❖ Clustered light gray pixels located in a specific area of the camera field of view.
- ❖ Non-clustered-light gray pixels located in a specific area of the camera field of view.
- ❖ Non-clustered-light gray pixels located anywhere in the camera field of view.

The calibration target can be:

- ❖ In the object to inspect.

- ❖ A specific object placed in the camera field of view during the calibration phase.

The appearance of the target must be:

- ❖ A neutral light gray color.
- ❖ The level of the brightest component within 75% to 90% of the full scale.
- ❖ The level of the darkest component above 15% of the full scale.

The target must contain at least 256 pixels satisfying the appearance requirements.

## Acquisition Channel Settings

The parameter `WBO_Mode` must be set to `ONCE`.

The parameters defining the position and the size of the `AWB_AREA` must be configured such that:

- ❖ It includes at least 256 pixels satisfying the calibration target appearance requirements.
- ❖ It contains at least 1 line and 32 columns of pixels.
- ❖ It is located entirely within the Camera Active Area.

Specifically on Grablink Base, Grablink DualBase, Grablink Full, and Grablink Full XR:

- ❖ The LUT Operator must be disabled.
- ❖ The position and the size of the cropping area must be configured such that it encompasses the `AWB_AREA`

### Related Links

**MultiCam reference topic(s)**

[WBO\\_Status](#) on page 50

## Automatic Calibration Timing

The color calibration process takes place during the first acquisition phase of a MultiCam acquisition sequence when the `WBO_Mode` is set to `ONCE`.

The White Balance Operator is disabled before the sequence starts.

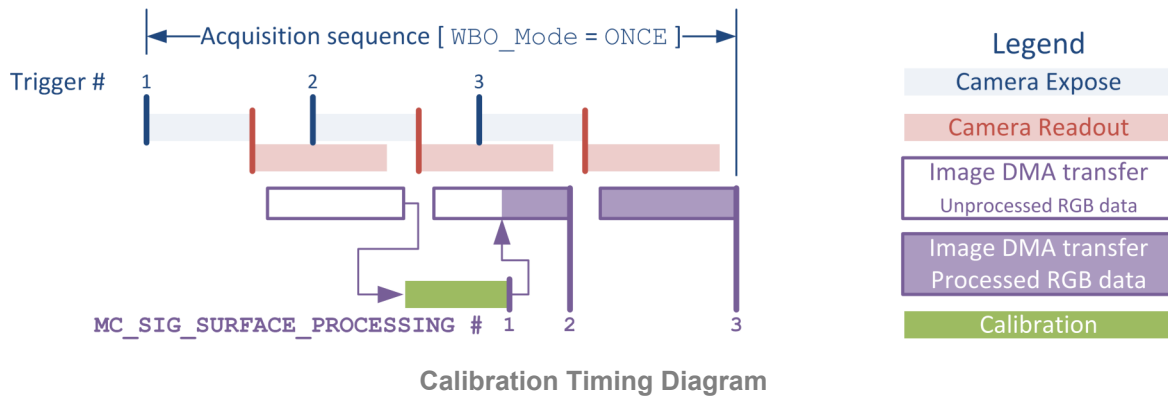
The calibration process begins when the DMA transfer of the first acquisition phase is completed. The first `MC_SIG_SURFACE_PROCESSING` signal of the sequence is delayed until the completion of the calibration process.

At the completion of a successful calibration process:

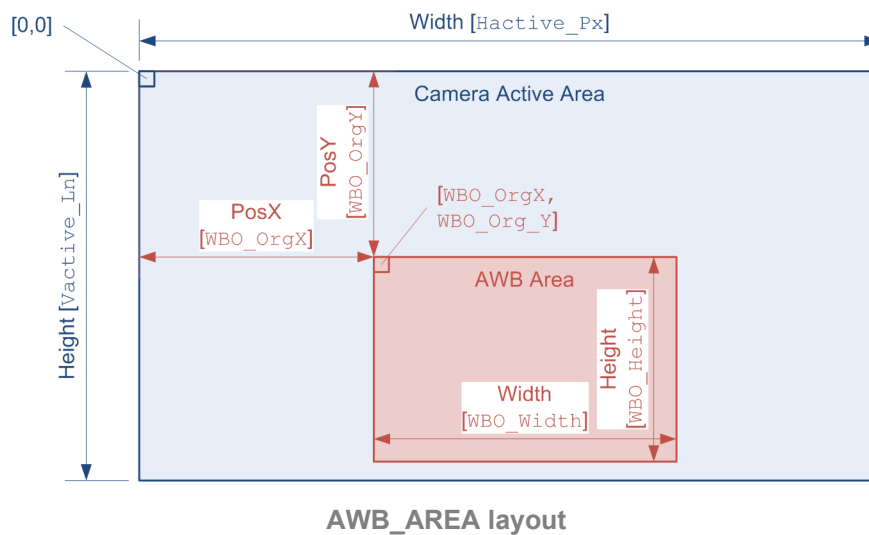
- ❖ The value of the parameter `WBO_Status` is set to `OK`.
- ❖ The values of parameters `WBO_GainR`, `WBO_GainG`, and `WBO_GainB` are updated with the calibration results.
- ❖ The White Balance Operator is reconfigured with the new settings.

At the completion of an unsuccessful calibration process:

- ❖ The value of the parameter `WBO_Status` is set to `NOT_OK`.
- ❖ The original values of parameters `WBO_GainR`, `WBO_GainG`, and `WBO_GainB` are restored.
- ❖ The White Balance Operator is reconfigured with the original settings.



## AWB\_AREA Settings Description



The `AWB_AREA` is a rectangular area within the Camera Active Window that is analyzed by the color balancing calibrator.

The size and the position of the `AWB_AREA` within the Camera Active Area is defined by the following parameters: `WBO_Width`, `WBO_Height`, `WBO_OrgX`, and `WBO_OrgY`.

The default size of the `AWB_AREA` is the whole Camera Active Area.

### Related Links

#### MultiCam reference topic(s)

[WBO\\_Width](#) on page 44

[WBO\\_Height](#) on page 45

[WBO\\_OrgX](#) on page 47

[WBO\\_OrgY](#) on page 48

# Image Geometrical Operators

## Image Cropping

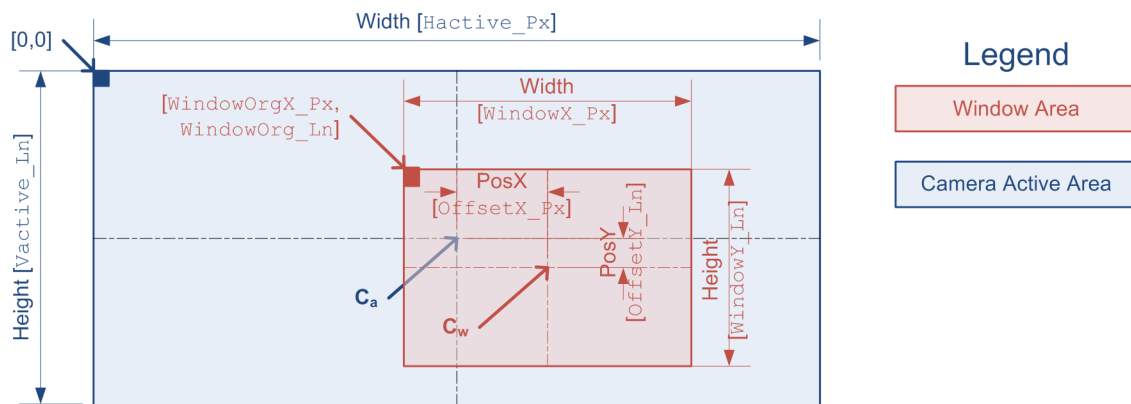
The image cropping operator - **ICO** - selects a subset of the pixels delivered by the camera to build the image delivered to the Host PC; this subset is named **Window Area**:

- ❖ For area-scan cameras, the **Window Area** is a single rectangular region of the 2D image sensor.
- ❖ For line-scan cameras, the **Window Area** is a single segment of the 1D image sensor.

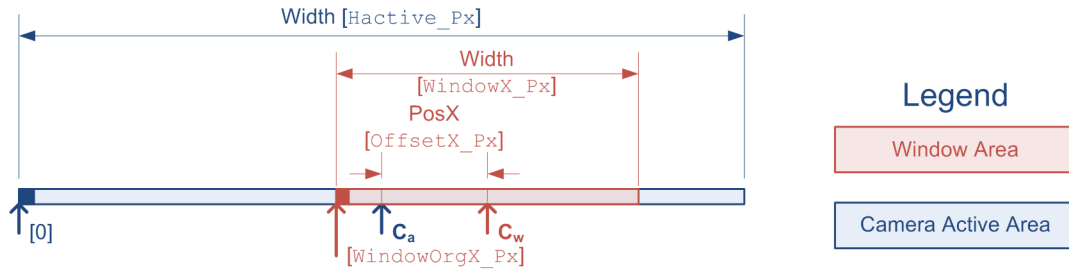
### ICO parameters

The ICO is controlled through the following Channel Class parameters of the Grabber Timing category:

- ❖ **GrabWindow**: the main control parameter.
- ❖ **WindowX\_Px**, **WindowY\_Ln**: integer parameters defining the size of the Window Area.
- ❖ **OffsetX\_Px**, and **OffsetY\_Ln**: integer parameters defining the position of the Window Area within the Camera Active Area.



Window Area Parameters for Area-Scan cameras



**Window Area Parameters for Line-Scan cameras**

**Note.** The position of the Window Area within the Camera Active Window is expressed as the difference of coordinates between  $C_w$ , the center of the Window Area, and  $C_a$ , the center of the Camera Active Area.

**Note.** The range of allowed values of  $OffsetX\_Px$  and  $OffsetY\_Ln$  parameters is automatically adjusted to force the Window Area to stay within the boundaries of the Camera Active Area.

**Note.**  $WindowOrgX\_Px$  reports the X-coordinate in the Camera Active Area of the leftmost pixels of the Window Area:  $WindowOrgX\_Px = (Hactive\_Px - WindowX\_Px) / 2$

**Note.** *For area-scan cameras only:*  $WindowOrgY\_Ln$  reports the Y-coordinate in the Camera Active Area of the topmost pixels of the Window Area:  $WindowOrgY\_Ln = (Vactive\_Ln - WindowY\_Ln) / 2$

## Configuring the Image Cropping Operator

By default, `GrabWindow` is set to `NOBLACK` disabling the ICO: the acquired image includes all the active pixels delivered by the camera without any surrounding weak or blind pixels on the image edges.

To enable image cropping, proceed as follows:

- ❖ Enable ICO by setting `GrabWindow` to `MAN`.
- ❖ Adjust the width of the Window Area using `WindowX_Px`. Any integer value ranging from **8** up to `Hactive_Px` is allowed.
- ❖ *For area-scan cameras only:* Adjust the height of the Window Area using `WindowY_Ln`. Any integer value ranging from **1** up to `Vactive_Ln` is allowed.
- ❖ Move horizontally the Window Area using `OffsetX_Px`. Increasing the value moves the Window Area towards the right of the Camera Active Area and vice-versa, decreasing the value moves the Window Area towards the left of the Camera Active Area.
- ❖ *For area-scan cameras only:* Move vertically the Window Area using `OffsetY_Ln`. Increasing the value moves the Window Area towards the top of the Camera Active Area and vice-versa, decreasing the value moves the Window Area towards the bottom of the Camera Active Area.

## Conditions of applicability

ICO is applicable to the following camera classes:

- ❖ Monochrome, RGB color, and Bayer CFA color **area-scan cameras**: any valid combination of `TapConfiguration` and `TapGeometry` is allowed except when `TapGeometry = *_2YE`
- ❖ Monochrome, and RGB color **line-scan cameras**.

### Related Links

#### MultiCam reference topic(s)

[GrabWindow](#) on page 28

[WindowX\\_Px](#) on page 29

[WindowY\\_Ln](#) on page 31

[OffsetX\\_Px](#) on page 32

[OffsetY\\_Ln](#) on page 34

[WindowOrgX\\_Px](#) on page 35

[WindowOrgY\\_Ln](#) on page 37

## Image Flipping Operators

The image flipping operator - **IFO** - performs mirroring of the image delivered to the Host PC:

- ❖ For area-scan cameras, the IFO is capable to perform left/right and top/bottom mirroring.
- ❖ For line-scan camera, the IFO is capable to perform left/right mirroring only.

The IFO is controlled through the following Channel Class parameters of the Cluster Category:

- ❖ `ImageFlipX` enables the left/right mirroring
- ❖ `ImageFlipY` enables the top/bottom mirroring.

By default, both operators are OFF disabling any mirroring.



### Conditions of applicability

IFO is applicable to the following camera classes:

- ❖ Monochrome, RGB color, and Bayer CFA color **area-scan cameras**
- ❖ Monochrome, and RGB color **line-scan cameras**

### Related Links

#### MultiCam reference topic(s)

[ImageFlipX](#) on page 53

[ImageFlipY](#) on page 54



# Metadata Insertion

## Short Description

---

When the *Metadata Insertion* feature is activated, the first 10 bytes of each image line are replaced by a fixed set of metadata including:

- ❖ The logical state of all (6) System I/O input lines
- ❖ The value of the motion encoder pulse counter
- ❖ The value of the Camera Link LVAL pulse counter

## Feature Availability and Requirements

The feature is available since version 6.9.1 of MultiCam on the following product:

- ❖ PC1626 Grablink Full XR

The feature is applicable to the following use cases:

| Use Case # | BoardTopology | Imaging | TapConfiguration | TapGeometry |
|------------|---------------|---------|------------------|-------------|
| 1          | MONO_DECA     | LINE    | DECA_10T8        | 1X10        |
| 2          | MONO_DECA     | TDI     | DECA_10T8        | 1X10        |

## Feature Control

The feature is controlled through the MultiCam parameter `MetadataInsertion`.

The feature is disabled by default. To activate the feature, set the parameter to `ENABLE` at the channel creation. The setting takes effect at the first channel activation.

### Related Links

**MultiCam reference topic(s)**

[MetadataInsertion](#) on page 26

## Functional Description

---

### Metadata Insertion

The first 10-byte parallel Camera Link video data of each acquired line are replaced by the following metadata:

| Camera Link Port Name | Metadata  |
|-----------------------|---|
| Port A                | Bits [3:0]: Logical state of the IIN[4:1] isolated input lines<br>Bits [5:4]: Logical state of the DIN[2:1] differential input lines<br>Bits [7:6]: Unused bits (logical state 0) |
| Port B                | Unused bits (logical state 0)   |
| Port C                | Motion Encoder Pulse counter value bits[7:0]  |
| Port D                | Motion Encoder Pulse counter value bits[15:8]   |
| Port E                | Motion Encoder Pulse counter value bits[23:16]  |
| Port F                | Motion Encoder Pulse counter value bits[31:24]  |
| Port G                | LVAL Pulse counter value bits[7:0]  |
| Port H                | LVAL Pulse counter value bits[15:8]   |
| Port I                | LVAL Pulse counter value bits[23:16]  |
| Port J                | LVAL Pulse counter value bits[31:24]  |

### Sampling Time

The state of System I/O input lines and the values of the counters are sampled at each rising edge of the Camera Link LVAL signal. The sampling time is NOT adjustable.

### System I/O Input State

The reported state is the logical state measured immediately after the electrical front-end.

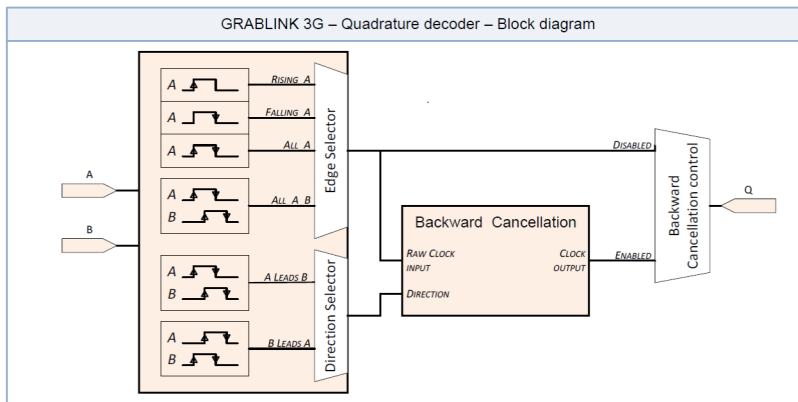
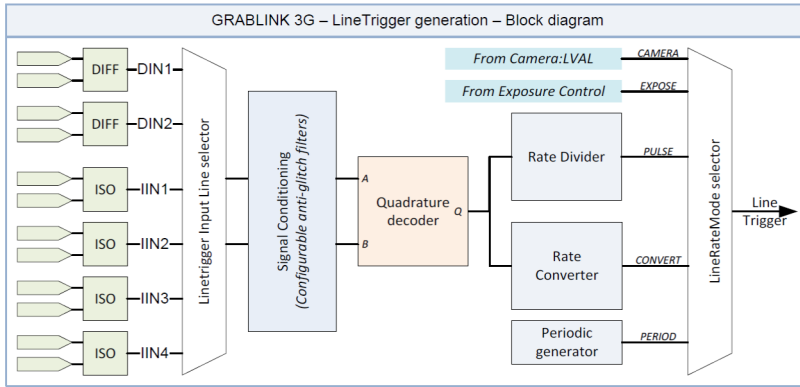
**Note.** There are no glitch removal filters, spurious state transitions may occur!

### Motion Encoder Pulse Counter

The 32-bit binary counter counts the pulses at the Q output of the Quadrature Decoder:







**Note.** Depending on the Quadrature Decoder settings, the counter increments by 0, 1, 2, or 4 units every encoder cycle.

The counter is not resettable:

- ❖ It is set to 0 at the driver initialization.
- ❖ It increments by 1 at every Q cycle.
- ❖ It wraps around to 0 when it reaches the maximum count 4,294,967,295 ( $=2^{32} - 1$ ).

## LVAL Pulse Counter

The 32-bit binary counter counts all the Camera Link LVAL pulses.

The counter is not resettable:

- ❖ It is set to 0 at the at the driver initialization.
- ❖ As soon as the Camera Link de-serializers are initialized, it increments by 1 at every LVAL cycle disregarding the acquisition conditions.
- ❖ It wraps around to 0 when it reaches the maximum count 4,294,967,295 ( $=2^{32} - 1$ ).

**Note.** The counter is incremented before its value is inserted as metadata: the first line cycle is marked 1.

# MultiCam Parameters Reference Update

## Channel Class - Grabber Configuration Category

---

### MetadataInsertion

| Property Name        | Property Value for MetadataInsertion Parameter |
|----------------------|--|
| Short description    | Controls metadata insertion into the image     |
| Class                | Channel  |
| Category             | Grabber Configuration                          |
| Level                | EXPERT   |
| Type                 | Enumerated                                     |
| Access               | Set and Get                                    |
| String identifier    | MetadataInsertion                              |
| C, C++ identifier    | MC_MetadataInsertion                           |
| Numerical identifier | 10842  |
| Available on         | Grablink Full XR                               |

### Parameter Description

This enumerated parameter controls the insertion of metadata into the image.

### Parameter Usage

#### Relevance condition(s):

The feature is applicable to the following use cases:

| Use Case # | BoardTopology | Imaging | TapConfiguration | TapGeometry |
|------------|---------------|---------|------------------|-------------|
| 1          | MONO_DECA     | LINE    | DECA_10T8        | 1X10        |
| 2          | MONO_DECA     | TDI     | DECA_10T8        | 1X10        |

The setting takes effect at the first channel activation.



## Enumerated Parameter Values

### Parameter Value ENABLE

|                          |                             |
|--------------------------|-----------------------------|
| <b>C, C++ identifier</b> | MC_MetadataInsertion_ENABLE |
| <b>Available on</b>      | Grablink Full XR            |

**Description:**

Enable insertion of metadata into the image.

### Parameter Value DISABLE

|                          |                              |
|--------------------------|------------------------------|
| <b>C, C++ identifier</b> | MC_MetadataInsertion_DISABLE |
| <b>Available on</b>      | Grablink Full XR             |

**Description:**

Disable insertion of metadata into the image. **Default value.**

**Related Links**

**Functional description topic(s)**

[Short Description](#) on page 23

## Channel Class - Grabber Timing Category

---

### GrabWindow

| Property Name        | Property Value for GrabWindow Parameter   |
|----------------------|---|
| Short description    | Method to define the grabbing window area |
| Class                | Channel                                   |
| Category             | Grabber Timing                            |
| Level                | ADJUST                                    |
| Type                 | Enumerated                                |
| Access               | Set and Get                               |
| String identifier    | GrabWindow                                |
| C, C++ identifier    | MC_GrabWindow                             |
| Numerical identifier | 683                                       |
| Available on         | All MultiCam boards                       |

### Parameter Description

This enumerated parameter selects the method defining the grabbing window area within the camera active area. For area-scan cameras, the grabbing window area is inferred from the camera active rectangular window. For line-scan cameras, the width of the grabbing window area is inferred from the camera active linear window.

### Enumerated Parameter Values

**Note.** Only the parameter values that are applicable to at least one product of the products selection are shown hereafter!

### Parameter Value NOBLACK

|                   |                       |
|-------------------|-----------------------|
| C, C++ identifier | MC_GrabWindow_NOBLACK |
| Available on      | All MultiCam boards   |

### Description:

For digital cameras, the grabbing window area matches exactly the Camera Active Area:

- ❖ Grabbing window area width = Hactive\_Px
- ❖ Grabbing window area height = Vactive\_Ln (*Area-scan cameras only!*)



- ❖ Grabbing window area X-position offset = 0
- ❖ Grabbing window area Y-position offset = 0 (*Area-scan cameras only!*)

**Default value.**

## Parameter Value MAN

|                          |                     |
|--------------------------|---------------------|
| <b>C, C++ identifier</b> | MC_GrabWindow_MAN   |
| <b>Available on</b>      | All MultiCam boards |

### Description:

When GrabWindow is set to MAN, the grabbing window area is defined by separate parameters:

- ❖ Grabbing window area width = WindowX\_Px
- ❖ Grabbing window area height = WindowY\_Ln (*Area-scan cameras only!*)
- ❖ Grabbing window area X-position offset = OffsetX\_Px
- ❖ Grabbing window area Y-position offset = OffsetY\_Ln (*Area-scan cameras only!*)

### Condition(s) of applicability:

- ❖ Line-scan cameras
- ❖ Area-scan cameras having a single region along the Y direction. For instance, the value is not applicable to cameras having a TapGeometry value suffixed \_2YE.

### Related Links

#### Functional description topic(s)

[Image Cropping](#) on page 20

## WindowX\_Px

| Property Name               | Property Value for WindowX_Px Parameter |
|-----------------------------|---|
| <b>Short description</b>    | Width of the grabbing window area       |
| <b>Class</b>                | Channel                                 |
| <b>Category</b>             | Grabber Timing                          |
| <b>Level</b>                | ADJUST                                  |
| <b>Type</b>                 | Integer                                 |
| <b>Access</b>               | Set and Get                             |
| <b>String identifier</b>    | WindowX_Px                              |
| <b>C, C++ identifier</b>    | MC_WindowX_Px                           |
| <b>Numerical identifier</b> | 826                                     |
| <b>Available on</b>         | All MultiCam boards                     |

## Parameter Description

This integer parameter reflects the width of the grabbing window area, expressed as a number of digitized pixels. The "get" value exactly reflects the actual window width. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.

## Parameter Usage

### Relevance condition(s):

- ❖ Manually defined grabbing window area (GrabWindow=MAN)

### Prerequisite action(s):

- ❖ Grabbing window definition method already selected through GrabWindow

**Directive:** Assigning a value smaller than Hactive\_Px enables the image cropping feature.

**Directive:** The grabbing window area must be included entirely within the camera active area.

## Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions   |
|-----------|--|
| 8         | <p><b>Applies to Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR.</b></p> <p><b>Description:</b><br/>The minimal width of the grabbing pixel area is 8 pixels.</p> |

| Max value | Description and applicability conditions  |
|-----------|---|
|           | <p><b>Description:</b><br/>The upper limit of the range is context dependent. The maximal width of the grabbing pixel area is equal to Hactive_Px.</p> <p><b>Default value.</b></p> |

## Related Links

### Functional description topic(s)

[Image Cropping](#) on page 20



## WindowY\_Ln

| Property Name        | Property Value for WindowY_Ln Parameter |
|----------------------|---|
| Short description    | Height of the grabbing window area      |
| Class                | Channel                                 |
| Category             | Grabber Timing                          |
| Level                | ADJUST                                  |
| Type                 | Integer                                 |
| Access               | Set and Get                             |
| String identifier    | WindowY_Ln                              |
| C, C++ identifier    | MC_WindowY_Ln                           |
| Numerical identifier | 827                                     |
| Available on         | All MultiCam boards                     |

### Parameter Description

This integer parameter reflects the height of the grabbing window area, expressed as a number of lines.

The "get" value exactly reflects the actual window height. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.

The parameter is available on all MultiCam products supporting area-scan cameras.

The parameter can be set when GrabWindow is set to MAN.

### Parameter Usage

#### Relevance condition(s):

- ❖ Manually defined grabbing window area (GrabWindow=MAN)
- ❖ Area-scan camera (Imaging = AREA) having a single region along the Y direction (TapGeometry ≠ \*\_2YE)

#### Prerequisite action(s):

- ❖ Grabbing window definition method already selected through GrabWindow

**Directive:** Assigning a value smaller than Vactive\_Ln enables the image cropping feature.

**Directive:** The grabbing window area must be included entirely within the camera active area.

### Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions  |
|-----------|---|
| 1         | <b>Applies to Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR.</b><br><b>Description:</b><br>The minimal height of the grabbing pixel area is 1 line. |

| Max value | Description and applicability conditions  |
|-----------|---|
|           | <b>Description:</b><br>The upper limit of the range is context dependent. The maximal width of the grabbing pixel area is equal to Vactive_Ln.<br><b>Default value.</b> |

#### Related Links

Functional description topic(s)

[Image Cropping](#) on page 20

## OffsetX\_Px

| Property Name        | Property Value for OffsetX_Px Parameter  |
|----------------------|--|
| Short description    | Horizontal position offset of the grabbing window area in the camera active area |
| Class                | Channel  |
| Category             | Grabber Timing   |
| Level                | ADJUST   |
| Type                 | Integer  |
| Access               | Set and Get  |
| String identifier    | OffsetX_Px   |
| C, C++ identifier    | MC_OffsetX_Px  |
| Numerical identifier | 825  |
| Available on         | All MultiCam boards  |

### Parameter Description

This integer parameter reflects the horizontal position offset of the center of the grabbing window area relative to the center of the camera active area.

The "get" value exactly reflects the shifted amount. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.



## Parameter Usage

### Relevance condition(s):

- ❖ Manually defined grabbing window area (GrabWindow=MAN)

### Prerequisite action(s):

- ❖ Grabbing window definition method already selected through GrabWindow
- ❖ Grabbing window height already set through WindowY\_Ln

**Directive:** A value of zero means that the grabbing window area is horizontally centered on the Camera Active Area. Increasing the value shifts the grabbing window area in the right direction. Decreasing the value shifts the grabbing window area in the left direction.

**Directive:** The grabbing window area must be included entirely within the camera active area.

## Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value    | Description and applicability conditions  |
|--------------|---|
|              | <p><b>Description:</b><br/>The grabbing window is at the leftmost position within the camera active area. The lower limit of the range is context dependent. The lowest value is equal to <math>[-(\text{Hactive\_Px} - \text{WindowX\_Px}) / 2]</math>.</p>    |
| Max value    | Description and applicability conditions  |
|              | <p><b>Description:</b><br/>The grabbing window is at the rightmost position within the camera active area. The upper limit of the range is context dependent. The upper value is equal to <math>[(\text{Hactive\_Px} - \text{WindowX\_Px} + 1) / 2]</math>.</p> |
| Other values | Description and applicability conditions  |
| 0            | <p><b>Description:</b><br/>The grabbing window area is horizontally centered on the grabbing window area.</p> <p><b>Default value.</b></p>  |

### Related Links

#### Functional description topic(s)

[Image Cropping](#) on page 20

## OffsetY\_Ln

| Property Name        | Property Value for OffsetY_Ln Parameter   |
|----------------------|---|
| Short description    | Vertical position offset of the grabbing window area in the camera active area. |
| Class                | Channel   |
| Category             | Grabber Timing  |
| Level                | ADJUST  |
| Type                 | Integer   |
| Access               | Set and Get   |
| String identifier    | OffsetY_Ln  |
| C, C++ identifier    | MC_OffsetY_Ln   |
| Numerical identifier | 686   |
| Available on         | All MultiCam boards   |

### Parameter Description

This integer parameter reflects the vertical position offset of the center of the Window Area relative to the center of the camera active area.

The "get" value exactly reflects the shifted amount. It may differ from the "set" value established by the user since MultiCam automatically corrects invalid values.

### Parameter Usage

#### Relevance condition(s):

- ❖ Manually defined grabbing window area (GrabWindow=MAN)
- ❖ Area-scan camera (Imaging = AREA) having a single region along the Y direction (TapGeometry ≠ \*\_2YE)

#### Prerequisite action(s):

- ❖ Grabbing window definition method already selected through GrabWindow
- ❖ Grabbing window height already set through WindowY\_Ln

**Directive:** Assigning a value of zero means that the grabbing window area is vertically centered on the Camera Active Area. Increasing the value shifts the grabbing window area in the downward direction. Decreasing the value shifts the grabbing window area in the upward direction.

**Directive:** The grabbing window area must be included entirely within the camera active area.

### Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.



**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions   |
|-----------|--|
|           | <p><b>Description:</b><br/>The grabbing window is at the uppermost position within the camera active area. The lower limit of the range is context dependent. The lowest value is equal to <math>[- (Vactive\_Ln - WindowY\_Ln) / 2]</math>.</p> |

| Max value | Description and applicability conditions  |
|-----------|---|
|           | <p><b>Description:</b><br/>The grabbing window is at the lowermost position within the camera active area. The upper limit of the range is context dependent. The upper value is equal to <math>[(Vactive\_Ln - WindowY\_Ln + 1) / 2]</math>.</p> |

| Other values | Description and applicability conditions   |
|--------------|--|
| 0            | <p><b>Description:</b><br/>The grabbing window area is vertically centered on the grabbing window area.</p> <p><b>Default value.</b></p> |

#### Related Links

##### Functional description topic(s)

[Image Cropping](#) on page 20

## WindowOrgX\_Px

| Property Name        | Property Value for WindowOrgX_Px Parameter   |
|----------------------|--|
| Short description    | X-coordinate of the upper-left corner of the grabbing window area  |
| Class                | Channel  |
| Category             | Grabber Timing   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Get Only   |
| String identifier    | WindowOrgX_Px  |
| C, C++ identifier    | MC_WindowOrgX_Px   |
| Numerical identifier | 8761   |
| Available on         | Grablink Express, Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

## Parameter Description

This integer parameter reports the X-coordinate, expressed as a number of pixels, of the upper left corner of the grabbing window area.

## Parameter Usage

### Relevance condition(s):

- ❖ Manually defined grabbing window area (GrabWindow=MAN)

## Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions  |
|-----------|---|
| 0         | <b>Description:</b><br>The X-coordinate of the upper left corner of the grabbing window area is 0 |

| Max value | Description and applicability conditions   |
|-----------|--|
|           | <b>Description:</b><br>The upper limit of the range is context dependent. The upper value is equal to [Hactive_Px - WindowX_Px]. |

## Related Links

### Functional description topic(s)

[Image Cropping](#) on page 20

## WindowOrgY\_Ln

| Property Name        | Property Value for WindowOrgY_Ln Parameter   |
|----------------------|--|
| Short description    | Y-coordinate of the upper-left corner of the grabbing window area  |
| Class                | Channel  |
| Category             | Grabber Timing   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Get Only   |
| String identifier    | WindowOrgY_Ln  |
| C, C++ identifier    | MC_WindowOrgY_Ln   |
| Numerical identifier | 8765   |
| Available on         | Grablink Express, Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This integer parameter reports the Y-coordinate, expressed as a number of lines, of the upper left corner of the grabbing window area.

### Parameter Usage

#### Relevance condition(s):

- ❖ Manually defined grabbing window area (GrabWindow=MAN)
- ❖ Area-scan camera (Imaging = AREA) having a single region along the Y direction (TapGeometry ≠ \*\_2YE)

### Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions  |
|-----------|---|
| 0         | <b>Description:</b><br>The Y-coordinate of the upper left corner of the grabbing window area is 0 |
| Max value | Description and applicability conditions  |
|           | <b>Description:</b><br>The upper limit of the range is context dependent.                         |

**Related Links**

Functional description topic(s)

[Image Cropping](#) on page 20

## Channel Class - White Balance Operator Category

---

### WBO\_Mode

| Property Name        | Property Value for WBO_Mode Parameter  |
|----------------------|--|
| Short description    | Operating mode of the white balance operator   |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | ADJUST   |
| Type                 | Enumerated   |
| Access               | Set and Get  |
| String identifier    | WBO_Mode   |
| C, C++ identifier    | MC_WBO_Mode  |
| Numerical identifier | 4715   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This enumerated parameter determines the operating mode of the White Balance Operator within a MultiCam acquisition sequence.

### Parameter Usage

**Relevance condition(s):**

- ❖ The camera is a color camera (Spectrum=COLOR).
- ❖ The acquisition channel delivers Y and/or RGB pixel data (ColorFormat ≠ BAYER\*)

### Enumerated Parameter Values

**Note.** Only the parameter values that are applicable to at least one product of the products selection are shown hereafter!



## Parameter Value NONE

|                          |  |
|--------------------------|--|
| <b>C, C++ identifier</b> | MC_WBO_Mode_NONE   |
| <b>Available on</b>      | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Description:

When WBO\_Mode is set to NONE, the White Balance Operator is disabled; the gain corrections are not applied. **Default value.**

## Parameter Value ONCE

|                          |  |
|--------------------------|--|
| <b>C, C++ identifier</b> | MC_WBO_Mode_ONCE   |
| <b>Available on</b>      | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Description:

When WBO\_Mode is set to ONCE, the image color balancing gains are automatically computed during the initial acquisition phase of every MultiCam acquisition sequence within the AWB\_AREA defined by parameters WBO\_OrgX, WBO\_OrgY, WBO\_Width, and WBO\_Height.

The parameters WBO\_GainR, WBO\_GainG, and WBO\_GainB are automatically set to the respective computed gain values.

All the delivered images of the acquisition sequence, including the first one, are properly color balanced.

### Description for Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR:

The White Balance Operator is disabled at the begin of the sequence and remains disabled until the occurrence of the first MC\_SIG\_SURFACE\_PROCESSING signal. The first delivered image is never color balanced; subsequent images remain partially or entirely unbalance until the White Balance Operator is configured.

## Parameter Value MANUAL

|                          |  |
|--------------------------|--|
| <b>C, C++ identifier</b> | MC_WBO_Mode_MANUAL   |
| <b>Available on</b>      | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Description:

When WBO\_Mode is set to MANUAL, the image color balance is performed with gains specified by parameters WBO\_GainR, WBO\_GainG and WBO\_GainB.

### Related Links

#### Functional description topic(s)

[White Balance Operator](#) on page 14

#### Functional description topic(s)

[What Is White Balance?](#) on page 16

## WBO\_GainR

| Property Name        | Property Value for WBO_GainR Parameter   |
|----------------------|--|
| Short description    | White balance correction factor for the red color component                                    |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Set and Get  |
| String identifier    | WBO_GainR  |
| C, C++ identifier    | MC_WBO_GainR   |
| Numerical identifier | 4717   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This integer parameter represents the correction factor applied by the White Balance Operator to the red color component.

The parameter values are expressed in 1/1000<sup>th</sup>. For instance a value of 1234 corresponds to a correction factor of 1.234.

### Parameter Usage

#### Relevance condition(s):

- ❖ Manually defined WBO gains (WBO\_Mode = MANUAL)

#### Prerequisite action(s):

- ❖ The WBO operation mode is already selected through WBO\_Mode

### Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.





**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions  |
|-----------|---|
| 1000      | <b>Description:</b><br>The gain correction factor is 1.000<br><b>Default value.</b> |

| Max value | Description and applicability conditions                   |
|-----------|--|
| 10000     | <b>Description:</b><br>The gain correction factor is 10.00 |

#### Related Links

##### Functional description topic(s)

[White Balance Operator](#) on page 14

## WBO\_GainG

| Property Name               | Property Value for WBO_GainG Parameter   |
|-----------------------------|--|
| <b>Short description</b>    | White balance correction factor for the green color component                                  |
| <b>Class</b>                | Channel  |
| <b>Category</b>             | White Balance Operator   |
| <b>Level</b>                | EXPERT   |
| <b>Type</b>                 | Integer  |
| <b>Access</b>               | Set and Get  |
| <b>String identifier</b>    | WBO_GainG  |
| <b>C, C++ identifier</b>    | MC_WBO_GainG   |
| <b>Numerical identifier</b> | 4719   |
| <b>Available on</b>         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This integer parameter represents the correction factor applied by the White Balance Operator to the green color component.

The parameter values are expressed in 1/1000<sup>th</sup>. For instance a value of 1234 corresponds to a correction factor of 1.234.

### Parameter Usage

**Relevance condition(s):**

- ❖ Manually defined WBO gains (WBO\_Mode = MANUAL)

#### Prerequisite action(s):

- ❖ The WBO operation mode is already selected through WBO\_Mode

## Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions  |
|-----------|---|
| 1000      | <b>Description:</b><br>The gain correction factor is 1.000<br><br><b>Default value.</b> |

| Max value | Description and applicability conditions                   |
|-----------|--|
| 10000     | <b>Description:</b><br>The gain correction factor is 10.00 |

#### Related Links

##### Functional description topic(s)

[White Balance Operator](#) on page 14

## WBO\_GainB

| Property Name        | Property Value for WBO_GainB Parameter   |
|----------------------|--|
| Short description    | White balance correction factor for the blue color component                                   |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Set and Get  |
| String identifier    | WBO_GainB  |
| C, C++ identifier    | MC_WBO_GainB   |
| Numerical identifier | 4720   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |



## Parameter Description

This integer parameter represents the correction factor applied by the White Balance Operator to the blue color component.

The parameter values are expressed in 1/1000<sup>th</sup>. For instance a value of 1234 corresponds to a correction factor of 1.234.

## Parameter Usage

### Relevance condition(s):

- ❖ Manually defined WBO gains (WBO\_Mode = MANUAL)

### Prerequisite action(s):

- ❖ The WBO operation mode is already selected through WBO\_Mode

## Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions  |
|-----------|---|
| 1000      | <b>Description:</b><br>The gain correction factor is 1.000<br><b>Default value.</b> |

| Max value | Description and applicability conditions                   |
|-----------|--|
| 10000     | <b>Description:</b><br>The gain correction factor is 10.00 |

## Related Links

### Functional description topic(s)

[White Balance Operator](#) on page 14

## WBO\_Width

| Property Name        | Property Value for WBO_Width Parameter   |
|----------------------|--|
| Short description    | Width of the Automatic White Balance Area  |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Set and Get  |
| String identifier    | WBO_Width  |
| C, C++ identifier    | MC_WBO_Width   |
| Numerical identifier | 5456   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This integer parameter represents the width, expressed as a number of pixels, of the rectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

### Parameter Usage

#### Relevance condition(s):

- ❖ Automatically defined WBO gains (WBO\_Mode = ONCE or CONTINUOUS)

#### Prerequisite action(s):

- ❖ The WBO operation mode is already selected through WBO\_Mode

**Directive:** The AWB\_AREA must include at least 256 pixels.

**Directive:** The AWB\_AREA must include at least 32 columns of pixels.

**Directive:** The AWB\_AREA must be included entirely within the camera active area.

**Directive:** The AWB\_AREA must be included entirely within the grabbing window area.

### Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions                               |
|-----------|--|
| 32        | <b>Description:</b><br>The minimal width of the AWB_AREA is 32 pixels. |

| Max value | Description and applicability conditions   |
|-----------|--|
|           | <b>Applies to Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR.</b><br><b>Description:</b><br>The upper limit of the range is context dependent.<br><b>Default value.</b> |

#### Related Links

##### Functional description topic(s)

[AWB\\_AREA Settings Description](#) on page 19

## WBO\_Height

| Property Name        | Property Value for WBO_Height Parameter  |
|----------------------|--|
| Short description    | Height of the Automatic White Balance Area   |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Set and Get  |
| String identifier    | WBO_Height   |
| C, C++ identifier    | MC_WBO_Height  |
| Numerical identifier | 5459   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This integer parameter represents the height, expressed as a number of lines, of the rectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

### Parameter Usage

**Relevance condition(s):**

- ❖ Automatically defined WBO gains (WBO\_Mode = ONCE or CONTINUOUS)

**Prerequisite action(s):**

- ❖ The WBO operation mode is already selected through WBO\_Mode

**Directive:** The AWB\_AREA must include at least 256 pixels.

**Directive:** The AWB\_AREA must include at least 1 line of pixels.

**Directive:** The AWB\_AREA must be included entirely within the camera active area.

**Directive:** The AWB\_AREA must be included entirely within the grabbing window area.

## Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions   |
|-----------|--|
| 1         | <p><b>Description:</b><br/>The minimal height of the AWB_AREA is 1 line.</p>   |
| Max value | Description and applicability conditions   |
|           | <p><b>Applies to Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR.</b></p> <p><b>Description:</b><br/>The upper limit of the range is context dependent.</p> <p><b>Default value.</b></p> |

### Related Links

**Functional description topic(s)**

[AWB\\_AREA Settings Description](#) on page 19



## WBO\_OrgX

| Property Name        | Property Value for WBO_OrgX Parameter  |
|----------------------|--|
| Short description    | X-coordinate of the upper-left corner of the AWB_AREA  |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Set and Get  |
| String identifier    | WBO_OrgX   |
| C, C++ identifier    | MC_WBO_OrgX  |
| Numerical identifier | 5449   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This integer parameter represents the X-coordinate, expressed as a number of pixels, of the upper left corner of a rectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

### Parameter Usage

#### Relevance condition(s):

- ❖ Automatically defined WBO gains (WBO\_Mode = ONCE or CONTINUOUS)

#### Prerequisite action(s):

- ❖ The WBO operation mode is already selected through WBO\_Mode

**Directive:** The AWB\_AREA must include at least 256 pixels.

**Directive:** The AWB\_AREA must include at least 32 columns of pixels.

**Directive:** The AWB\_AREA must be included entirely within the camera active area.

**Directive:** The AWB\_AREA must be included entirely within the grabbing window area.

### Integer Parameter Values Range

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions  |
|-----------|---|
| 0         | <b>Description:</b><br>The X-coordinate of the upper-left corner of the AWB_AREA is 0 |

| Max value | Description and applicability conditions                                  |
|-----------|---|
|           | <b>Description:</b><br>The upper limit of the range is context dependent. |

#### Related Links

##### Functional description topic(s)

[AWB\\_AREA Settings Description](#) on page 19

## WBO\_OrgY

| Property Name        | Property Value for WBO_OrgY Parameter  |
|----------------------|--|
| Short description    | Y-coordinate of the upper-left corner of the AWB_AREA  |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | EXPERT   |
| Type                 | Integer  |
| Access               | Set and Get  |
| String identifier    | WBO_OrgY   |
| C, C++ identifier    | MC_WBO_OrgY  |
| Numerical identifier | 5452   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This integer parameter represents the Y-coordinate, expressed as a number of lines, of the upper left corner of a rectangular region within the camera active area that is used by the Automatic White Balance feature to compute the white balance correction factors.

### Parameter Usage

#### Relevance condition(s):

- ❖ Automatically defined WBO gains (WBO\_Mode = ONCE or CONTINUOUS)





**Prerequisite action(s):**

- ❖ The WBO operation mode is already selected through WBO\_Mode

**Directive:** The AWB\_AREA must include at least 256 pixels.

**Directive:** The AWB\_AREA must include at least 1 line of pixels.

**Directive:** The AWB\_AREA must be included entirely within the camera active area.

**Directive:** The AWB\_AREA must be included entirely within the grabbing window area.

**Integer Parameter Values Range**

Unless otherwise specified, any integer value within the [Min-Value, Max\_value] range boundaries is allowed.

**Note.** Only the range characteristic values that are applicable to at least one product of the products selection are shown hereafter!

| Min value | Description and applicability conditions   |
|-----------|--|
| 0         | <p><b>Description:</b><br/>The Y-coordinate of the upper-left corner of the AWB_AREA is 0</p> <p><b>Default value.</b></p> |

| Max value | Description and applicability conditions  |
|-----------|---|
|           | <p><b>Description:</b><br/>The upper limit of the range is context dependent.</p> |

**Related Links****Functional description topic(s)**

[AWB\\_AREA Settings Description](#) on page 19

## WBO\_Status

| Property Name        | Property Value for WBO_Status Parameter  |
|----------------------|--|
| Short description    | Status of the automatic white balance learning block   |
| Class                | Channel  |
| Category             | White Balance Operator   |
| Level                | EXPERT   |
| Type                 | Enumerated   |
| Access               | Get Only   |
| String identifier    | WBO_Status   |
| C, C++ identifier    | MC_WBO_Status  |
| Numerical identifier | 8940   |
| Available on         | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This enumerated parameter shows the result status of the automatic white balance computation.

### Parameter Usage

#### Relevance condition(s):

- ❖ Automatically defined WBO gains (WBO\_Mode = ONCE or CONTINUOUS)

#### Prerequisite action(s):

- ❖ The WBO operation mode is already selected through WBO\_Mode
- ❖ At least one acquisition phase already completed.

### Enumerated Parameter Values

**Note.** Only the parameter values that are applicable to at least one product of the products selection are shown hereafter!

### Parameter Value OK

|                   |  |
|-------------------|--|
| C, C++ identifier | MC_WBO_Status_OK   |
| Available on      | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

#### Description:



The automatic white balance learning block succeeds to balance the color. The white balance color gain settings are updated.

## Parameter Value NOT\_OK

|                          |  |
|--------------------------|--|
| <b>C, C++ identifier</b> | MC_WBO_Status_NOT_OK   |
| <b>Available on</b>      | Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Description:

The automatic white balance learning block fails to balance the color. The white balance color gain settings are not updated.

### Related Links

#### Functional description topic(s)

[Automatic Calibration Requirements](#) on page 17

## Channel Class - Cluster Category

---

### MaxFillingSurfaces

| Property Name               | Property Value for MaxFillingSurfaces Parameter                   |
|-----------------------------|---|
| <b>Short description</b>    | Filling surfaces control  |
| <b>Class</b>                | Channel   |
| <b>Category</b>             | Cluster   |
| <b>Level</b>                | EXPERT  |
| <b>Type</b>                 | Enumerated  |
| <b>Access</b>               | Set and Get   |
| <b>String identifier</b>    | MaxFillingSurfaces  |
| <b>C, C++ identifier</b>    | MC_MaxFillingSurfaces   |
| <b>Numerical identifier</b> | 10712   |
| <b>Available on</b>         | Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Parameter Description

This parameter specifies the operation of the cluster mechanism regarding the number of surfaces it is allowed to put in the FILLING state.

## Parameter Usage

### Prerequisite action(s):

- ❖ The parameter must be set prior to the channel activation, i.e. when ChannelState=IDLE

**Directive:** Allocate a sufficient amount of surfaces and manage the surfaces such that the cluster mechanism maintains a sufficient amount of surfaces in the MC\_SurfaceState\_FILLING state to cover the largest system interrupt latencies.

## Enumerated Parameter Values

**Note.** Only the parameter values that are applicable to at least one product of the products selection are shown hereafter!

### Parameter Value MINIMUM

|                          |   |
|--------------------------|---|
| <b>C, C++ identifier</b> | MC_MaxFillingSurfaces_MINIMUM                                     |
| <b>Available on</b>      | Grablinc Base, Grablinc DualBase, Grablinc Full, Grablinc Full XR |

#### Description:

The cluster mechanism is allowed to put only one surface in the FILLING state at a time.

### Parameter Value MAXIMUM

|                          |   |
|--------------------------|---|
| <b>C, C++ identifier</b> | MC_MaxFillingSurfaces_MAXIMUM                                     |
| <b>Available on</b>      | Grablinc Base, Grablinc DualBase, Grablinc Full, Grablinc Full XR |

#### Description:

The cluster mechanism is allowed to put up to 512 surfaces in the FILLING state at a time.**Default value.**

#### Related Links

##### Functional description topic(s)

[Cluster Mechanism](#) on page 7



## ImageFlipX

| Property Name        | Property Value for ImageFlipX Parameter |
|----------------------|---|
| Short description    | Horizontal mirroring effect             |
| Class                | Channel                                 |
| Category             | Cluster                                 |
| Level                | ADJUST                                  |
| Type                 | Enumerated                              |
| Access               | Set and Get                             |
| String identifier    | ImageFlipX                              |
| C, C++ identifier    | MC_ImageFlipX                           |
| Numerical identifier | 1340                                    |
| Available on         | All MultiCam boards                     |

### Parameter Description

The horizontal mirroring effect can be thought as turning the image around a vertical axis (first column becomes last column).

### Parameter Usage

### Enumerated Parameter Values

**Note.** Only the parameter values that are applicable to at least one product of the products selection are shown hereafter!

### Parameter Value OFF

|                   |                     |
|-------------------|---------------------|
| C, C++ identifier | MC_ImageFlipX_OFF   |
| Available on      | All MultiCam boards |

#### Description:

No horizontal mirroring effect.

## Parameter Value ON

|                          |  |
|--------------------------|--|
| <b>C, C++ identifier</b> | MC_ImageFlipX_ON   |
| <b>Available on</b>      | Piccolo Alert, Piccolo Diligent, Domino Iota, Domino Alpha 2, Domino Melody, Domino Harmony, Domino Symphony, Grablink Value, Grablink Expert 2, Grablink Quickpack ColorScan, Grablink Avenue, Grablink Express, Grablink Quickpack CFA PCIe, Grablink Base, Grablink DualBase, Grablink Full, Grablink Full XR |

### Description:

Horizontal mirror applied.

### Related Links

#### Functional description topic(s)

[Image Flipping Operators](#) on page 22

## ImageFlipY

| Property Name               | Property Value for ImageFlipY Parameter |
|-----------------------------|---|
| <b>Short description</b>    | Vertical mirroring effect               |
| <b>Class</b>                | Channel                                 |
| <b>Category</b>             | Cluster                                 |
| <b>Level</b>                | ADJUST                                  |
| <b>Type</b>                 | Enumerated                              |
| <b>Access</b>               | Set and Get                             |
| <b>String identifier</b>    | ImageFlipY                              |
| <b>C, C++ identifier</b>    | MC_ImageFlipY                           |
| <b>Numerical identifier</b> | 525                                     |
| <b>Available on</b>         | All MultiCam boards                     |

### Parameter Description

The vertical mirroring effect can be thought as turning the image around a horizontal axis (first line becomes last line).

### Parameter Usage

### Enumerated Parameter Values

**Note.** Only the parameter values that are applicable to at least one product of the products selection are shown hereafter!



## Parameter Value OFF

|                          |                     |
|--------------------------|---------------------|
| <b>C, C++ identifier</b> | MC_ImageFlipY_OFF   |
| <b>Available on</b>      | All MultiCam boards |

**Description:**

No vertical mirroring effect.

## Parameter Value ON

|                          |                     |
|--------------------------|---------------------|
| <b>C, C++ identifier</b> | MC_ImageFlipY_ON    |
| <b>Available on</b>      | All MultiCam boards |

**Description:**

Vertical mirror applied.

**Related Links****Functional description topic(s)**

[Image Flipping Operators](#) on page 22

# Surface Class - Surface Dynamics Category

---

## SurfaceState

| Property Name               | Property Value for SurfaceState Parameter |
|-----------------------------|---|
| <b>Short description</b>    | State of the surface                      |
| <b>Class</b>                | Surface                                   |
| <b>Category</b>             | Surface Dynamics                          |
| <b>Level</b>                | ADJUST                                    |
| <b>Type</b>                 | Enumerated                                |
| <b>Access</b>               | Set and Get                               |
| <b>String identifier</b>    | SurfaceState                              |
| <b>C, C++ identifier</b>    | MC_SurfaceState                           |
| <b>Numerical identifier</b> | 31  |
| <b>Available on</b>         | All MultiCam boards                       |

## Parameter Description

This parameter allows to read and write the current state of the surface.

## Enumerated Parameter Values

**Note.** Only the parameter values that are applicable to at least one product of the products selection are shown hereafter!

### Parameter Value FREE

|                          |                      |
|--------------------------|----------------------|
| <b>C, C++ identifier</b> | MC_SurfaceState_FREE |
| <b>Available on</b>      | All MultiCam boards  |

#### Description:

The surface is able to receive image data from the grabber.

### Parameter Value FILLING

|                          |                         |
|--------------------------|-------------------------|
| <b>C, C++ identifier</b> | MC_SurfaceState_FILLING |
| <b>Available on</b>      | All MultiCam boards     |

#### Description:

The surface is currently receiving or ready to receive image data from the grabber.

### Parameter Value FILLED

|                          |                        |
|--------------------------|------------------------|
| <b>C, C++ identifier</b> | MC_SurfaceState_FILLED |
| <b>Available on</b>      | All MultiCam boards    |

#### Description:

The surface has finished receiving image data from the grabber, and thus is ready for processing

### Parameter Value PROCESSING

|                          |                            |
|--------------------------|----------------------------|
| <b>C, C++ identifier</b> | MC_SurfaceState_PROCESSING |
| <b>Available on</b>      | All MultiCam boards        |

#### Description:

The surface is being processed by the host processor.





## Parameter Value RESERVED

|                          |                          |
|--------------------------|--------------------------|
| <b>C, C++ identifier</b> | MC_SurfaceState_RESERVED |
| <b>Available on</b>      | All MultiCam boards      |

**Description:**

The surface is removed from the standard state transition.

**Related Links****Functional description topic(s)**

[Cluster Mechanism](#) on page 7