

# Writing Applications with xiAPI

## Default parameters

After camera is opened by xiOpenDevice the default camera parameters are set by API. The default parameters might be different in different API versions. In order to ensure that your application will have camera in expected state with any API version - please set all parameters expected by your application to required value.

---

## AE Setup

### XI\_PRM\_EXP\_PRIORITY or "exp\_priority"

**Description:** Exposure priority for Auto Exposure / Auto Gain function.

Value: **1.0** >>> meaning: **Exposure priority. Only exposure will be changed.**

Value: **0.5** >>> meaning: **Exposure and gain will be used (50%:50%).**

Value: **0.0** >>> meaning: **Gain priority. Only gain will be changed.**

**Type:** Float.

**Default value:** 1.0

**Typical range:** [ 0.0, 1.0 ]

**Usage:**

```
float value = 0.0;
```

```
xiGetParamFloat(handle, XI_PRM_EXP_PRIORITY, &value);
```

```
xiSetParamFloat(handle, XI_PRM_EXP_PRIORITY, value);
```

### XI\_PRM\_AG\_MAX\_LIMIT or "ag\_max\_limit"

**Description:** Maximum limit of gain in AEAG procedure.

**Type:** Float.

**Default value:** Depends on camera type (dB).

**Usage:**

```
float value = 0.0;
```

```
xiGetParamFloat(handle, XI_PRM_AG_MAX_LIMIT, &value);
```

```
xiSetParamFloat(handle, XI_PRM_AG_MAX_LIMIT, value);
```

### XI\_PRM\_AE\_MAX\_LIMIT or "ae\_max\_limit"

**Description:** Maximum limit of exposure (in uSec) in AEAG procedure.

**Type:** Integer.

**Default value:** 200000

**Typical range:** [ 0, 1000000 ]

**Usage:**

```
int value = 0;
```

```
xiGetParamInt(handle, XI_PRM_AE_MAX_LIMIT, &value);
```

```
xiSetParamInt(handle, XI_PRM_AE_MAX_LIMIT, value);
```

## XI\_PRM\_AEAG\_LEVEL or "aeag\_level"

**Description:** Average intensity of output signal AEAG should achieve(in %).

**Type:** Integer.

**Default value:** 50

**Typical range:** [ 0, 100 ]

**Usage:**

```
int value = 0;
```

```
xiGetParamInt(handle, XI_PRM_AEAG_LEVEL, &value);
```

```
xiSetParamInt(handle, XI_PRM_AEAG_LEVEL, value);
```

---

## Performance

### XI\_PRM\_LIMIT\_BANDWIDTH or "limit\_bandwidth"

Description: Camera acquisition data-rate Limit on transport layer in Megabits (1000000) per second. API controls the camera clock or increases the line period by 1 in order to achieve the closest data-rate as the Limit value set, ensuring the data-rate is below the Limit. This parameter can be used to decrease data-rate e.g. when more cameras are connected to same interface to share same channel. In order to activate the limit - application should set also [XI\\_PRM\\_LIMIT\\_BANDWIDTH\\_MODE = XI\\_ON](#), see example below.

**Note:** Controlling method (clock or line period) depends on the camera model.

**Type:** Integer.

**Default value:** 0

**Usage:**

```
xiSetParamInt(handle, XI_PRM_LIMIT_BANDWIDTH, datarate_in_Mbits_sec);
```

Example: See more at our [application note about Multiple Cameras Setup](#).

```
// get interface data rate
int interface_data_rate_mbps=2500;
// calculate data rate for each camera
#define CONNECTED_CAMERAS_TO_SAME_HUB 3
int camera_data_rate = interface_data_rate_mbps / CONNECTED_CAMERAS_TO_SAME_HUB;
#define MARGIN_MBitsPER_SECOND 300
camera_data_rate -= MARGIN_MBitsPER_SECOND;
// set data rate
xiSetParamInt(handle, XI_PRM_LIMIT_BANDWIDTH, camera_data_rate);
// enable the limiting
xiSetParamInt(handle, XI_PRM_LIMIT_BANDWIDTH_MODE, XI_ON);
```

## XI\_PRM\_LIMIT\_BANDWIDTH\_MODE or "limit\_bandwidth\_mode"

**Description:** Controls if the `XI_PRM_LIMIT_BANDWIDTH` is active. When disabled, lower level specific features are expected to control the throughput. When enabled, `XI_PRM_LIMIT_BANDWIDTH` controls the overall throughput.

**Note:** This parameter is not supported on MQ, MU, MD, MR camera families.

**Type:** Enumerator.

**Default value:** XI\_ON

**Usage:**

```
int limit_bandwidth_mode = 0;
xiGetParamInt(handle, XI_PRM_LIMIT_BANDWIDTH_MODE, &limit_bandwidth_mode);
xiSetParamInt(handle, XI_PRM_LIMIT_BANDWIDTH_MODE, XI_OFF);
```

Value	Description
XI_OFF	Turn parameter off
XI_ON	Turn parameter on

## XI\_PRM\_SENSOR\_DATA\_BIT\_DEPTH or "sensor\_bit\_depth"

**Description:** Returns the bit depth of the pixel data received from sensor.

**Note:** [Read more at XiAPI Image Data Flow.](#)

**Type:** Enumerator.

**Default value:** 0

**Is invalidated by:** `XI_PRM_IMAGE_DATA_FORMAT`, `XI_PRM_USER_SET_LOAD`, `XI_PRM_DUAL_ADC_MODE`, `XI_PRM_DOWNSAMPLING`

**Usage:**

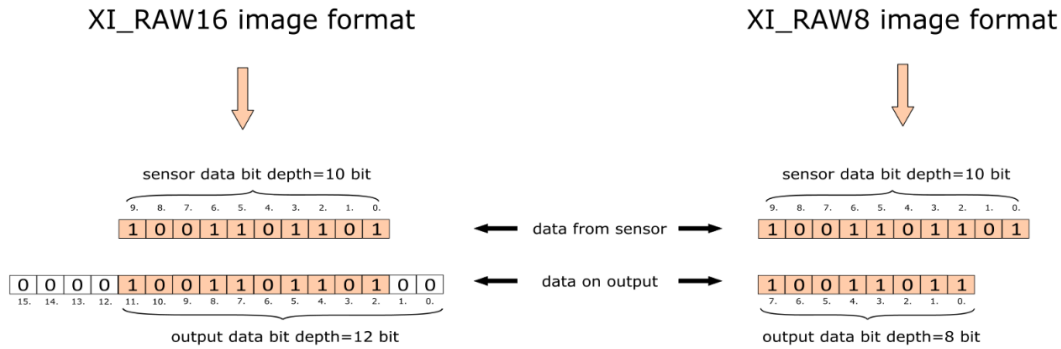
```
int sensor_bit_depth = 0;
xiGetParamInt(handle, XI_PRM_SENSOR_DATA_BIT_DEPTH, &sensor_bit_depth);
xiSetParamInt(handle, XI_PRM_SENSOR_DATA_BIT_DEPTH, XI_BPP_8);
```

Value	Description
XI_BPP_8	8 bit per pixel
XI_BPP_9	9 bit per pixel
XI_BPP_10	10 bit per pixel
XI_BPP_11	11 bit per pixel
XI_BPP_12	12 bit per pixel
XI_BPP_13	13 bit per pixel
XI_BPP_14	14 bit per pixel
XI_BPP_15	15 bit per pixel
XI_BPP_16	16 bit per pixel
XI_BPP_24	24 bit per pixel
XI_BPP_32	32 bit per pixel

XI\_PRM\_OUTPUT\_DATA\_BIT\_DEPTH or "output\_bit\_depth"

**Description:** The bit depth of the output data from camera (=transport layer).

Note: Read more at [XiAPI Image Data Flow](#).



**Type:** Enumerator.

**Default value:** 0

**Is invalidated by:** XI\_PRM\_IMAGE\_DATA\_FORMAT, XI\_PRM\_DP\_PARAM\_VALUE, XI\_PRM\_BINNING\_VERTICAL, XI\_PRM\_BINNING\_HORIZONTAL, XI\_PRM\_DECIMATION\_VERTICAL, XI\_PRM\_DECIMATION\_HORIZONTAL, XI\_PRM\_SHUTTER\_TYPE

**Usage:**

```
int output_bit_depth = 0;
xiGetParamInt(handle, XI_PRM_OUTPUT_DATA_BIT_DEPTH, &output_bit_depth);
xiSetParamInt(handle, XI_PRM_OUTPUT_DATA_BIT_DEPTH, XI_BPP_8);
```

Value	Description
XI_BPP_8	8 bit per pixel
XI_BPP_9	9 bit per pixel
XI_BPP_10	10 bit per pixel
XI_BPP_11	11 bit per pixel
XI_BPP_12	12 bit per pixel
XI_BPP_13	13 bit per pixel
XI_BPP_14	14 bit per pixel
XI_BPP_15	15 bit per pixel
XI_BPP_16	16 bit per pixel
XI_BPP_24	24 bit per pixel
XI_BPP_32	32 bit per pixel

XI\_PRM\_IMAGE\_DATA\_BIT\_DEPTH or "image\_data\_bit\_depth"

**Description:** The bit depth of the pixel data returned by function xiGetImage. If MONO16 or RAW16 image formats are used this parameter defines the alignment of the data on the xiGetImage.

**Type:** Enumerator.

**Default value:** 0

Is invalidated by: [XI\\_PRM\\_IMAGE\\_DATA\\_FORMAT](#)

**Usage:**

```
int image_data_bit_depth = 0;
xiGetParamInt(handle, XI_PRM_IMAGE_DATA_BIT_DEPTH, &image_data_bit_depth);
xiSetParamInt(handle, XI_PRM_IMAGE_DATA_BIT_DEPTH, XI_BPP_8);
```

Value	Description
XI_BPP_8	8 bit per pixel
XI_BPP_9	9 bit per pixel
XI_BPP_10	10 bit per pixel
XI_BPP_11	11 bit per pixel
XI_BPP_12	12 bit per pixel
XI_BPP_13	13 bit per pixel
XI_BPP_14	14 bit per pixel
XI_BPP_15	15 bit per pixel
XI_BPP_16	16 bit per pixel
XI_BPP_24	24 bit per pixel
XI_BPP_32	32 bit per pixel

## XI\_PRM\_OUTPUT\_DATA\_PACKING or "output\_bit\_packing"

Description: This feature enables bit packing on transport data layer, thus increasing the maximum frame rate when data with 10 or 12 bits per pixel is transported. For more info please see [Transport Data Packing feature description](#).

Note: Read more at [XiAPI Image Data Flow](#).

**Type:** Integer.

**Default value:** XI\_OFF

**Is invalidated by:** [XI\\_PRM\\_IMAGE\\_DATA\\_FORMAT](#), [XI\\_PRM\\_OUTPUT\\_DATA\\_BIT\\_DEPTH](#), [XI\\_PRM\\_DP\\_PARAM\\_VALUE](#), [XI\\_PRM\\_BINNING\\_VERTICAL](#), [XI\\_PRM\\_BINNING\\_HORIZONTAL](#), [XI\\_PRM\\_DECIMATION\\_VERTICAL](#), [XI\\_PRM\\_DECIMATION\\_HORIZONTAL](#), [XI\\_PRM\\_SHUTTER\\_TYPE](#)

**Usage:**

```
int value = 0;
xiGetParamInt(handle, XI_PRM_OUTPUT_DATA_PACKING, &value);
xiSetParamInt(handle, XI_PRM_OUTPUT_DATA_PACKING, XI_ON);
```

## XI\_PRM\_OUTPUT\_DATA\_PACKING\_TYPE or "output\_bit\_packing\_type"

Description: This feature chooses output data packing type(ximea grouping 10g160, 12g192, 14g224), PFNC packing 10p, 12p...). For more info please see [Transport Data Packing feature description](#).

**Type:** Enumerator.

**Default value:** XI\_DATA\_PACK\_XI\_GROUPING

**Usage:**

```
int output_bit_packing_type = 0;  
xiGetParamInt(handle, XI_PRM_OUTPUT_DATA_PACKING_TYPE, &output_bit_packing_type);  
xiSetParamInt(handle, XI_PRM_OUTPUT_DATA_PACKING_TYPE, XI_DATA_PACK_XI_GROUPING);
```

Value	Description
XI_DATA_PACK_XI_GROUPING	Data grouping (10g160, 12g192, 14g224).
XI_DATA_PACK_PFNC_LSB_PACKING	Data packing (10p, 12p)