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.

Debounce Setup

XI_PRM_DEBOUNCE_T0 or "dbnc_t0"

Description: Debounce time (x * 10us) for transition to inactive level of GPI selected by XI_PRM_DEBOUNCE_POL.

Type: Integer.

Default value: 0

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_DEBOUNCE_T0, &value);
xiSetParamInt(handle, XI_PRM_DEBOUNCE_T0, value);
```

XI_PRM_DEBOUNCE_T1 or "dbnc_t1"

Description: Debounce time (x * 10us)for transition to active level of GPI selected by XI_PRM_DEBOUNCE_POL

Type: Integer.

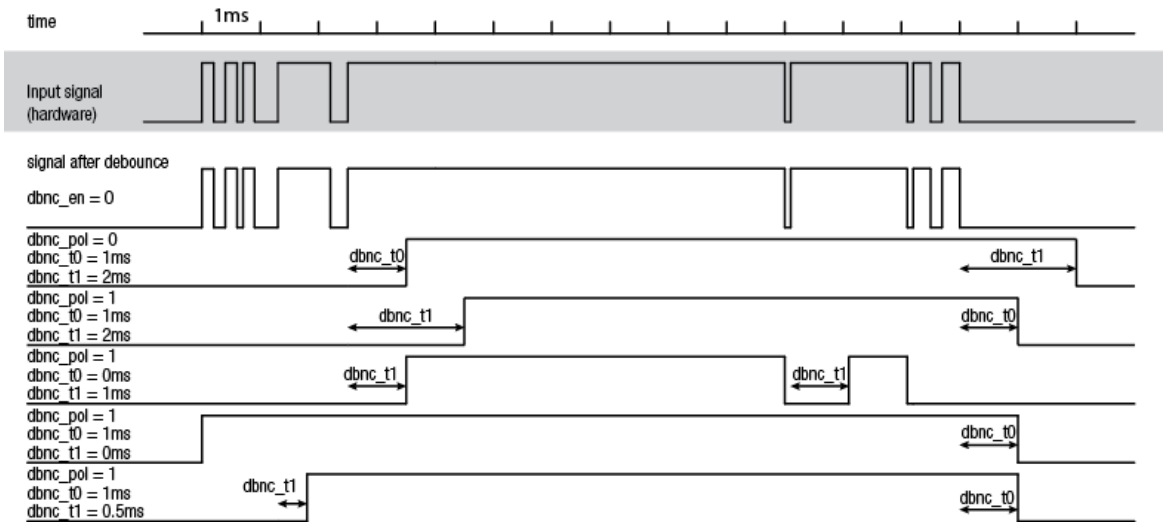
Default value: 0

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_DEBOUNCE_T1, &value);
xiSetParamInt(handle, XI_PRM_DEBOUNCE_T1, value);
```

XI_PRM_DEBOUNCE_POL or "dbnc_pol"

Description: Debounce polarity selects active level of GPI (see XI_PRM_GPI_SELECTOR parameter). Does not inverts the signal if set.



Type: Integer.

Default value: 0

Typical range: [0, 1]

Usage:

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

Lens Control

Note: Some of XIMEA cameras can be equipped with controlled lens. API for lens control contains couple of parameters.

Lens tested OK with CB cameras:

CANON EF 50mm f/1.4 USM

CANON EF 50mm f/1.8 II

CANON EF 24-105 f4 L IS USM

CANON EF 17-40mm f/4L USM

CANON EF 100mm f/2.8 Macro USM

CANON EF-S 17-55mm f/2.8 IS USM

CANON EF 70-200mm f/4L IS USM

CANON EF 50mm f/1.8 STM

CANON EF-S 24mm f/2.8 STM

CANON EF-S 10-18mm f/4.5-5.6 IS STM

CANON EF-S 18-135mm f/3.5-5.6 IS STM

Canon EF 200mm f/2.8L II USM

Canon EF 180mm f/3.5L Macro USM

Sigma 150mm f/2.8 EX DG OS HSM APO Macro

Sigma 15mm f/2.8 EX DG

XI_PRM_LENS_MODE or "lens_mode"

Description: Status of lens control interface. This shall be set to XI_ON before any Lens operations.

Type: Integer.
Default value: XI_OFF
Usage:

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

XI_PRM_LENS_APERTURE_VALUE or "lens_aperture_value"
Description: Current lens aperture value in aperture stops. Examples: 2.8, 4, 5.6, 8, 11.
Type: Float.
Default value: 1.0
Usage:

```
float value = 0.0;
xiGetParamFloat(handle, XI_PRM_LENS_APERTURE_VALUE, &value);
xiSetParamFloat(handle, XI_PRM_LENS_APERTURE_VALUE, value);
```

XI_PRM_LENS_APERTURE_INDEX or "lens_aperture_index"
Description: Current lens aperture motor step value.
Type: Integer.
Default value: 1
Usage:

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

XI_PRM_LENS_FOCUS_MOVEMENT_VALUE or "lens_focus_movement_value"

Description: Lens current focus movement value to be used by XI_PRM_LENS_FOCUS_MOVE in motor steps. Positive numbers will direct the movement to infinity. Negative numbers will direct the movement to macro.

Type: Integer.
Default value: 0
Usage:

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

XI_PRM_LENS_FOCUS_MOVE or "lens_focus_move"

Description: Moves lens focus motor by steps set in XI_PRM_LENS_FOCUS_MOVEMENT_VALUE

Type: Integer.

Default value: 0

Usage:

```
xiSetParamInt(handle, XI_PRM_LENS_MODE, XI_ON);
xiSetParamFloat(handle, XI_PRM_LENS_FOCUS_MOVEMENT_VALUE, 10);
xiSetParamInt(handle, XI_PRM_LENS_FOCUS_MOVE, 0); // move 10 steps to infinity
```

XI_PRM_LENS_FOCAL_LENGTH or "lens_focal_length"

Description: Lens focal distance in mm. This parameter is constant for prime lens and can change in real time for zoom lens.

Type: Float.

Default value: 1.0

Usage:

```
float zoom_min_mm = 0;
float zoom_max_mm = 0;
xiGetParamFloat(handle, XI_PRM_LENS_FOCAL_LENGTH XI_PRM_INFO_MIN, &zoom_min_mm);
xiGetParamFloat(handle, XI_PRM_LENS_FOCAL_LENGTH XI_PRM_INFO_MAX, &zoom_max_mm);
```

XI_PRM_LENS_FEATURE_SELECTOR or "lens_feature_selector"

Description: Selects the current feature which is accessible by XI_PRM_LENS_FEATURE

Type: Enumerator.

Default value: XI_LENS_FEATURE_MOTORIZED_FOCUS_SWITCH

Usage:

See parameter XI_PRM_LENS_FEATURE

Value	Description
XI_LENS_FEATURE_MOTORIZED_FOCUS_SWITCH	Status of lens motorized focus switch
XI_LENS_FEATURE_MOTORIZED_FOCUS_BOUNDED	On read = 1 if motorized focus is on one of limits.
XI_LENS_FEATURE_MOTORIZED_FOCUS_CALIBRATION	(planned feature) On read = 1 if motorized focus is calibrated. Write 1 to start calibration.
XI_LENS_FEATURE_IMAGE_STABILIZATION_ENABLED	On read = 1 if image stabilization is enabled. Write 1 to enable image stabilization.
XI_LENS_FEATURE_IMAGE_STABILIZATION_SWITCH_STATUS	On read = 1 if image stabilization switch is in position On.
XI_LENS_FEATURE_IMAGE_ZOOM_SUPPORTED	On read = 1 if lens supports zoom = are not prime.

XI_PRM_LENS_FEATURE or "lens_feature"

Description: Allows access to lens feature value currently selected by XI_PRM_LENS_FEATURE_SELECTOR.

Type: Float.

Default value: 0.0

Usage:

```
xiSetParamInt(handle, XI_PRM_LENS_FEATURE_SELECTOR,
XI_LENS_FEATURE_MOTORIZED_FOCUS_SWITCH);

int switch_status = 0;
xiGetParamInt(handle, XI_PRM_LENS_FEATURE, &witch_status);
if (switch_status > 0)
   printf("The motorized focus switch on the lens is switched on.\n");
```

Device info parameters

XI_PRM_DEVICE_NAME or "device_name"

Description: Return device name.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_DEVICE_NAME, &value, sizeof(value));
```

XI_PRM_DEVICE_TYPE or "device_type"

Description: Returns device type (1394, USB2.0, USB3.0, PCIe, ...).

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_DEVICE_TYPE, &value, sizeof(value));
```

XI_PRM_DEVICE_MODEL_ID or "device_model_id"

Description: Returns the device model id.

Type: Integer.

Default value: 0

Usage:

```
int value = 0;
```

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

XI_PRM_SENSOR_MODEL_ID or "sensor_model_id"

Description: Returns the device sensor model id.

Type: Integer.

Default value: 0

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_SENSOR_MODEL_ID, &value);
```

XI_PRM_DEVICE_SN or "device_sn"

Description: Returns device serial number. Only string form is possible. It might contain also alphabet characters.

Type: String.

Default value: 0

Usage:

```
char sn[100] = "";
xiGetParamString(handle, XI_PRM_DEVICE_SN, sn, sizeof(sn));
```

Example:

```
// show serial number of this camera
char sn[100] = "";
xiGetParamString(handle, XI_PRM_DEVICE_SN, sn, sizeof(sn));
printf("Serial number of the camera is: %s\n",sn);
```

XI_PRM_DEVICE_SENS_SN or "device_sens_sn"

Description: Returns sensor serial number.

Type: String.

Default value: 0

Usage:

```
char sens_sn[100] = "";
xiGetParamString(handle, XI_PRM_DEVICE_SENS_SN, sens_sn, 100);
```

XI_PRM_DEVICE_INSTANCE_PATH or "device_inst_path"

Description: Returns device instance path in operating system.

Type: String.

Default value: -

Usage:

```
char path[100] = "";
```

```
xiGetParamString(handle, XI_PRM_DEVICE_INSTANCE_PATH, path, sizeof(path));
```

XI_PRM_DEVICE_LOCATION_PATH or "device_loc_path"

Description: Returns device location path in operating system. It should reflect the connection position.

Type: String.

Default value: -

Usage:

```
char path[100] = "";
xiGetParamString(handle, XI_PRM_DEVICE_LOCATION_PATH, path, sizeof(path));
```

XI_PRM_DEVICE_USER_ID or "device_user_id"

Description: Get/Set custom user ID stored in camera non volatile memory. This can be later used as a handle for opening or identification.

Note1: It is currently available only on some models

Note2: For xiQ camera devices are supported maximum length 56 characters.

Note3: For xiC camera devices are supported maximum length 48 characters.

Note4: For xiB, xiT, xiX camera devices are supported maximum length 4 characters. (Power off/on required after User ID changed)

Type: String.

Default value: -

Usage:

```
char name[100] = "";
xiGetParamString(handle,XI_PRM_DEVICE_USER_ID,name,sizeof(name));
```

Example: Example of setting and retrieving the DEVICE_USER_ID

Only after a successful reboot of the camera, it can be opened with the newly assigned

XI_PRM_DEVICE_MANIFEST or "device_manifest"

Description: Get XML of current device parameters and capabilities.

Note: Available currently on xiQ USB3.0, xiB, xiT cameras.

Type: String.

Default value: -

Usage:

```
char* manifest_data = NULL;
#define MANIF_MAX_SIZE 2*1024*1024
manifest_data = malloc(MANIF_MAX_SIZE);
xiGetParamString(handle,XI_PRM_DEVICE_MANIFEST,manifest_data,MANIF_MAX_SIZE);
```

XI_PRM_IMAGE_USER_DATA or "image_user_data"

Description: Sets the user data (32bit number) into camera. The following frame captured by camera will have this number stored at image header. The number is accessible later after xiGetImage in XI_IMG structure as image_user_data.

Supported cameras: xiB, xiT

Type: Integer.

Default value: 0

Typical range: [0, 0xFFFFFFFF]

Usage:

```
uint32_t data = 7;
xiSetParamInt(handle,XI_PRM_IMAGE_USER_DATA,data);
xiGetImage(handle,5000,&image);
printf("Image captured has user_data:%d\n",image.image_user_data);
```

Device acquisition settings

XI_PRM_IMAGE_DATA_FORMAT_RGB32_ALPHA or "imgdataformatrgb32alpha"

Description: The alpha channel of RGB32 output image format(see XI_PRM_IMAGE_DATA_FORMAT).

Type: Integer.

Default value: 0

Typical range: [0, 0xFFFF]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_IMAGE_DATA_FORMAT_RGB32_ALPHA, &value);
xiSetParamInt(handle, XI_PRM_IMAGE_DATA_FORMAT_RGB32_ALPHA, value);
```

XI_PRM_IMAGE_PAYLOAD_SIZE or "imgpayloadsize"

Description: Buffer size in bytes sufficient for output image returned by GetImage

Type: Integer.

Default value: 0

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_IMAGE_PAYLOAD_SIZE, &value);
```

XI_PRM_TRANSPORT_PIXEL_FORMAT or "transport_pixel_format"

Description: Transport pixel format is format of data transported by link to transport layer. It might be modified after setting of XI_PRM_IMAGE_DATA_FORMAT, XI_PRM_OUTPUT_DATA_PACKING, XI_PRM_OUTPUT_DATA_BIT_DEPTH, ...

Type: Enumerator.

Default value: XI_GenTL_Image_Format_Mono8

Usage:

```
int transport_pixel_format = 0;
xiGetParamInt(handle, XI_PRM_TRANSPORT_PIXEL_FORMAT, &transport_pixel_format);
xiSetParamInt(handle, XI_PRM_TRANSPORT_PIXEL_FORMAT, XI_GenTL_Image_Format_Mono8);
```

XI_PRM_TRANSPORT_DATA_TARGET or "transport_data_target"

Description: Sets image data delivery target to CPU RAM (default) or GPU RAM.

How to configure GPUDirect for memory transfers

How to configure CUDA for memory transfers

Type: Enumerator.

Default value: XI_TRANSPORT_DATA_TARGET_CPU_RAM

Usage:

```
int transport_data_target = 0;
xiGetParamInt(handle, XI_PRM_TRANSPORT_DATA_TARGET, &transport_data_target);
xiSetParamInt(handle, XI_PRM_TRANSPORT_DATA_TARGET, XI_TRANSPORT_DATA_TARGET_CPU_RAM);
```

Value	Description
XI_TRANSPORT_DATA_TARGET_CPU_RAM	normal CPU memory buffer is used for image data
XI_TRANSPORT_DATA_TARGET_GPU_RAM	data is delivered straight to GPU memory using GPUDirect technology
XI_TRANSPORT_DATA_TARGET_UNIFIED	CUDA managed memory is used for image data.
XI_TRANSPORT_DATA_TARGET_ZEROCOPY	CUDA zerocopy memory is used for image data.

XI_PRM_SENSOR_CLOCK_FREQ_HZ or "sensor_clock_freq_hz"

Description: Set or return the sensor clock frequency. This clock is specific to sensor used. See documentation/application for the camera to use this parameter.

Type: Float.

Default value: Depends on sensor model.

Is invalidated by: XI_PRM_LIMIT_BANDWIDTH

Usage:

```
float value = 0.0;
```

xiGetParamFloat(handle, XI_PRM_SENSOR_CLOCK_FREQ_HZ, &value);
xiSetParamFloat(handle, XI_PRM_SENSOR_CLOCK_FREQ_HZ, value);

XI_PRM_SENSOR_CLOCK_FREQ_INDEX or "sensor_clock_freq_index"
Description: Sensor clock frequency. Selects frequency on cameras which supports only some specific frequencies.

Type: Integer.
Default value: Depends on sensor model.
Usage:

int value = 0;
xiGetParamInt(handle, XI_PRM_SENSOR_CLOCK_FREQ_INDEX, &value);
xiSetParamInt(handle, XI_PRM_SENSOR_CLOCK_FREQ_INDEX, value);

XI_PRM_SENSOR_OUTPUT_CHANNEL_COUNT or "sensor_output_channel_count"

Description: Number of output channels from sensor used for data transfer.
Type: Enumerator.
Default value: Depends on sensor model.
Usage:

int sensor_output_channel_count = 0;
xiGetParamInt(handle, XI_PRM_SENSOR_OUTPUT_CHANNEL_COUNT, &sensor_output_channel_count);

xiSetParamInt(handle, XI_PRM_SENSOR_OUTPUT_CHANNEL_COUNT, XI_CHANN_CNT2);

Value	Description
XI_CHANN_CNT2	2 sensor readout channels.
XI_CHANN_CNT4	4 sensor readout channels.
XI_CHANN_CNT8	8 sensor readout channels.
XI_CHANN_CNT16	16 sensor readout channels.
XI_CHANN_CNT24	24 sensor readout channels.
XI_CHANN_CNT32	32 sensor readout channels.
XI_CHANN_CNT48	48 sensor readout channels.

XI_PRM_FRAMERATE or "framerate"

Description: Defines frames per second of sensor. See more details in article Frame Rate Control On some camera models it is possible to change or limit acquisition frame rate. Frame rate value should be within possible range, use XI_PRM_INFO_MAX, XI_PRM_INFO_MIN.

Note1: Use following code to set the frame rate to 10 FPS on MQ, MD cameras.

xiSetParamInt(handle, XI_PRM_ACQ_TIMING_MODE, XI_ACQ_TIMING_MODE_FRAME_RATE);

xiSetParamFloat(handle, XI_PRM_FRAMERATE, 10);

Note2: Use following code to limit the frame rate to 10 FPS on CB,MT,MX,MC cameras.

```
xiSetParamInt(handle, XI_PRM_ACQ_TIMING_MODE, XI_ACQ_TIMING_MODE_FRAME_RATE_LIMIT);
xiSetParamFloat(handle, XI_PRM_FRAMERATE, 10);
```

Type: Float.

Default value: 0.0

Is invalidated by: XI_PRM_IMAGE_DATA_FORMAT, XI_PRM_EXPOSURE, XI_PRM_ACQ_TIMING_MODE, XI_PRM_LIMIT_BANDWIDTH, XI_PRM_LIMIT_BANDWIDTH_MODE, XI_PRM_DOWNSAMPLING_TYPE, XI_PRM_DOWNSAMPLING, XI_PRM_BINNING_VERTICAL, XI_PRM_BINNING_HORIZONTAL, XI_PRM_DECIMATION_VERTICAL, XI_PRM_DECIMATION_HORIZONTAL, XI_PRM_WIDTH, XI_PRM_HEIGHT, XI_PRM_OUTPUT_DATA_PACKING, XI_PRM_OUTPUT_DATA_PACKING_TYPE, XI_PRM_SENSOR_DATA_BIT_DEPTH, XI_PRM_OUTPUT_DATA_BIT_DEPTH, XI_PRM_DUAL_ADC_MODE, XI_PRM_SENSOR_FEATURE_VALUE

Usage:

```
float value = 0.0;
xiGetParamFloat(handle, XI_PRM_FRAMERATE, &value);
xiSetParamFloat(handle, XI_PRM_FRAMERATE, value);
```

XI_PRM_COUNTER_SELECTOR or "counter_selector"

Description: Selects which frame counter must be returned

Note1: It returns number of skipped frames on the transport layer, number of skipped frames on API layer, number of successfully transferred frames.

Type: Enumerator.

Default value: XI_CNT_SEL_TRANSPORT_SKIPPED_FRAMES

Usage:

```
int counter_selector = 0;
xiGetParamInt(handle, XI_PRM_COUNTER_SELECTOR, &counter_selector);
xiSetParamInt(handle, XI_PRM_COUNTER_SELECTOR, XI_CNT_SEL_TRANSPORT_SKIPPED_FRAMES);
```

Value	Description
XI_CNT_SEL_TRANSPORT_SKIPPED_FRAMES	Number of skipped frames on transport layer (e.g. when image gets lost while transmission). Occur when capacity of transport channel does not allow to transfer all data.
XI_CNT_SEL_API_SKIPPED_FRAMES	Number of skipped frames on API layer. Occur when application does not process the images as quick as they are received from the camera.
XI_CNT_SEL_TRANSPORT_TRANSFERRED_FRAMES	Number of delivered buffers since last acquisition start.

XI_CNT_SEL_FRAME_MISSED_TRIGGER_DUETO_OVERLAP	Number of missed triggers overlapped with exposure or read-out stage of previous frame – see XI_PRM_TRG_OVERLAP. (see Note1)
XI_CNT_SEL_FRAME_MISSED_TRIGGER_DUETO_FRAME_BUFFER_OVR	Number of missed triggers due to frame buffer full. (see Note1)
XI_CNT_SEL_FRAME_BUFFER_OVERFLOW	Internal camera frame buffer memory (RAM) full events counter. It can be incremented multiple times per one frame. (see Note1)
XI_CNT_SEL_TRANSPORT_QUEUE_UNDERRUN	Incremented when camera starts to transfer new image, however no target buffer is queued in the transport queue. Connected to GenTL.STREAM_INFO_NUM_UNDERRUN. (see Note1)
XI_CNT_SEL_ACQUISITION_AUTO_RESTARTED_ON_FAILURE	Acquisition can be restarted, due to failures on bus

Note1: Available only on cameras series: xiX, xiB, xiT, xiC, xiMUdeveloped since 2023 (MU196, MU050, MU051).

XI_PRM_COUNTER_VALUE or "counter_value"

Description: Returns value of selected (by XI_PRM_COUNTER_SELECTOR) frame counter.

Note: All counters are reset with the camera open, and counters XI_CNT_SEL_TRANSPORT_SKIPPED_FRAMES, XI_CNT_SEL_API_SKIPPED_FRAMES and XI_CNT_SEL_TRANSPORT_TRANSFERRED_FRAMES are also reset with acquisition start.

Type: Integer.

Default value: 0

Usage:

```
int number_of_skipped_frames = 0;
xiSetParamInt(handle, XI_PRM_COUNTER_SELECTOR, XI_CNT_SEL_API_SKIPPED_FRAMES);
xiGetParamInt(handle, XI_PRM_COUNTER_VALUE, &number_of_skipped_frames);
```

XI_PRM_ACQ_TIMING_MODE or "acq_timing_mode"

Description: This parameter defines the acquisition timing mode. More information about enumerators XI_ACQ_TIMING_MODE_FRAME_RATE and XI_ACQ_TIMING_MODE_FRAME_RATE_LIMIT please refer to our Frame Rate Control support page.

Type: Enumerator.

Default value: XI_ACQ_TIMING_MODE_FREE_RUN

Usage:

```
int acq_timing_mode = 0;
xiGetParamInt(handle, XI_PRM_ACQ_TIMING_MODE, &acq_timing_mode);
xiSetParamInt(handle, XI_PRM_ACQ_TIMING_MODE, XI_ACQ_TIMING_MODE_FREE_RUN);
```

Value	Description

XI_ACQ_TIMING_MODE_FREE_RUN	camera acquires images at a maximum possible framerate
XI_ACQ_TIMING_MODE_FRAME_RATE	Selects a mode when sensor frame acquisition frequency is set to parameter FRAMERATE
XI_ACQ_TIMING_MODE_FRAME_RATE_LIMIT	Selects a mode when sensor frame acquisition frequency is limited by parameter FRAMERATE

XI_PRM_AVAILABLE_BANDWIDTH or "available_bandwidth"

Description: Measure available interface bandwidth. Unit is Megabits (1000000) per sec.

Note: Some parameters could be changed by getting available bandwidth. Please set camera parameters to needed value after getting of available bandwidth.

Type: Integer.

Default value: 0

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_AVAILABLE_BANDWIDTH, &value);
```

XI_PRM_BUFFER_POLICY or "buffer_policy"

Description: Defines buffer handling. Can be safe, data will be copied to user/app buffer or unsafe, user will get internally allocated buffer without data copy. Size of the image buffer can be obtained by parameter XI_PRM_IMAGE_PAYLOAD_SIZE

Note: Click to below link to open simple description of buffer policy. buffer_policy_in_xiApi.png

Type: Enumerator.

Default value: XI_BP_UNSAFE

Usage:

```
int buffer_policy = 0;
xiGetParamInt(handle, XI_PRM_BUFFER_POLICY, &buffer_policy);
xiSetParamInt(handle, XI_PRM_BUFFER_POLICY, XI_BP_UNSAFE);
```

Value	Description
XI_BP_UNSAFE	User gets pointer to internally allocated circle buffer and data may be overwritten by device.
XI_BP_SAFE	Data from device will be copied to user allocated buffer or xiApi allocated memory.

XI_PRM_LUT_EN or "LUTEnable"

Description: Activates Look-Up-Table (LUT).

Note1: Possible value: 0 - sensor pixels are transferred directly

Note2: Possible value: 1 - sensor pixels are mapped through LUT

Note3: LUT parameters are valid only for some cameras. E.g. xiQ supports LUT. xiMU (MU9PM-MH) does NOT support it.

Note4: For xiQ cameras setting XI_PRM_LUT_EN also uploads previously set values in to camera. Values are latched in API.

Type: Integer.

Default value: XI_OFF

Usage:

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

XI_PRM_LUT_INDEX or "LUTIndex"

Description: Controls the index (offset) of the coefficient to access in the LUT.

Note1: All xiQ cameras have LUT N-bit to N-bit, based on the XI_PRM_SENSOR_DATA_BIT_DEPTH For the specific camera. All xiC/xiX/xiT cameras have LUT 12-bit to 12-bit.

Note2: Range of applicable indexes depends on sensor digitization bit depth (sensor_bit_depth). Use XI_PRM_INFO_MAX, XI_PRM_INFO_MIN.

Type: Integer.

Default value: 0

Usage:

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

XI_PRM_LUT_VALUE or "LUTValue"

Description: Defines value at entry LUTIndex of the LUT.

Note1: Range of applicable values depends on sensor digitization bit depth (sensor_bit_depth). Use XI_PRM_INFO_MAX, XI_PRM_INFO_MIN.

Note2: All xiQ cameras have LUT N-bit to N-bit, based on the XI_PRM_SENSOR_DATA_BIT_DEPTH of the specific camera. All xiC/xiX/xiT cameras have LUT 12-bit to 12-bit.

Note2: For xiQ cameras setting values has no direct effect on image, only after setting XI_PRM_LUT_EN to value 1, will apply all changes to camera.

Type: Integer.

Default value: 0

Usage:

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

XI_PRM_TRG_DELAY or "trigger_delay"

Description: When set delay time is inserted between camera trigger input and activating sensor integration. Delay time is set in us.

Note: Setting of this parameter is applicable for selected cameras:

xiX, xiB, xiT, xiC

xiMU (MU9). Granularity of real delay duration depends on sensor settings (line read out time). Typical

Type: Integer.

Default value: 0

Is invalidated by: XI_PRM_TRG_SELECTOR

Usage:

```
xiSetParamInt(handle, XI_PRM_TRG_SELECTOR, XI_TRG_SEL_FRAME_START);
xiSetParamInt(handle, XI_PRM_TRG_DELAY, 5000);  //5ms delay
```

XI_PRM_TS_RST_MODE or "ts_rst_mode"

Description: Defines way timestamp reset engine is armed.

Type: Enumerator.

Default value: XI_TS_RST_ARM_ONCE

Usage:

```
xiSetParamInt(handle, XI_PRM_TS_RST_MODE, XI_TS_RST_ARM_ONCE);
xiSetParamInt(handle, XI_PRM_TS_RST_SOURCE, XI_TS_RST_SRC_TRIGGER);
```

Value	Description
XI_TS_RST_ARM_ONCE	Engine is disabled after TimeStamp has been reset after selected event.
XI_TS_RST_ARM_PERSIST	Engine is armed permanently so each selected event will trigger TimeStamp reset.

XI_PRM_TS_RST_SOURCE or "ts_rst_source"

Description: Defines source for timestamp reset engine as well as the polarity active signal. The engine is edge sensitive.

Note: Number of active GPI or GPO depends on camera model.

Type: Enumerator.

Default value: XI_TS_RST_OFF

Usage:

```
xiSetParamInt(handle, XI_PRM_TS_RST_MODE, XI_TS_RST_ARM_ONCE);
xiSetParamInt(handle, XI_PRM_TS_RST_SOURCE, XI_TS_RST_SRC_TRIGGER);
```

Value	Description
XI_TS_RST_OFF	No source selected TimeStamp reset is not armed.
XI_TS_RST_SRC_GPI_1	GPI1 rising edge is active (signal after de-bounce module)
XI_TS_RST_SRC_GPI_2	GPI2 rising edge is active
XI_TS_RST_SRC_GPI_3	GPI3 rising edge is active
XI_TS_RST_SRC_GPI_4	GPI4 rising edge is active
XI_TS_RST_SRC_GPI_1_INV	GPI1 falling edge is active
XI_TS_RST_SRC_GPI_2_INV	GPI2 falling edge is active
XI_TS_RST_SRC_GPI_3_INV	GPI3 falling edge is active

XI_TS_RST_SRC_GPI_4_INV	GPI4 falling edge is active
XI_TS_RST_SRC_GPO_1	TimeStamp reset source selected GPO1
XI_TS_RST_SRC_GPO_2	TimeStamp reset source selected GPO2
XI_TS_RST_SRC_GPO_3	TimeStamp reset source selected GPO3
XI_TS_RST_SRC_GPO_4	TimeStamp reset source selected GPO4
XI_TS_RST_SRC_GPO_1_INV	TimeStamp reset source selected GPO1 inverted
XI_TS_RST_SRC_GPO_2_INV	TimeStamp reset source selected GPO2 inverted
XI_TS_RST_SRC_GPO_3_INV	TimeStamp reset source selected GPO3 inverted
XI_TS_RST_SRC_GPO_4_INV	TimeStamp reset source selected GPO4 inverted
XI_TS_RST_SRC_TRIGGER	TRIGGER to sensor rising edge is active
XI_TS_RST_SRC_TRIGGER_INV	TRIGGER to sensor rising edge is active
XI_TS_RST_SRC_SW	TRIGGER to sensor rising edge is active. TimeStamp is reset by software take effect imminently.
XI_TS_RST_SRC_EXPACTIVE	Exposure Active signal rising edge
XI_TS_RST_SRC_EXPACTIVE_INV	Exposure Active signal falling edge
XI_TS_RST_SRC_FVAL	Frame valid signal rising edge (internal signal in camera)
XI_TS_RST_SRC_FVAL_INV	Frame valid signal falling edge (internal signal in camera)
XI_TS_RST_SRC_GPI_5	GPI5 rising edge is active
XI_TS_RST_SRC_GPI_6	GPI6 rising edge is active
XI_TS_RST_SRC_GPI_5_INV	GPI5 falling edge is active
XI_TS_RST_SRC_GPI_6_INV	GPI6 falling edge is active
XI_TS_RST_SRC_GPI_7	TimeStamp reset source selected GPI7 (after de bounce)
XI_TS_RST_SRC_GPI_8	TimeStamp reset source selected GPI8 (after de bounce)
XI_TS_RST_SRC_GPI_9	TimeStamp reset source selected GPI9 (after de bounce)
XI_TS_RST_SRC_GPI_10	TimeStamp reset source selected GPI10 (after de bounce)
XI_TS_RST_SRC_GPI_11	TimeStamp reset source selected GPI11 (after de bounce)
XI_TS_RST_SRC_GPI_7_INV	TimeStamp reset source selected GPI7 inverted (after de bounce)
XI_TS_RST_SRC_GPI_8_INV	TimeStamp reset source selected GPI8 inverted (after de bounce)
XI_TS_RST_SRC_GPI_9_INV	TimeStamp reset source selected GPI9 inverted (after de bounce)
XI_TS_RST_SRC_GPI_10_INV	TimeStamp reset source selected GPI10 inverted (after de bounce)
XI_TS_RST_SRC_GPI_11_INV	TimeStamp reset source selected GPI11 inverted (after de bounce)