

# 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.

---

## APIContextControl

### XI\_PRM\_API\_CONTEXT\_LIST or "xiapi\_context\_list"

**Description:** API Context contains the text representation of current settings for offline image processing. It can be gotten while acquisition to store the context. Respectively, it can be set while offline processing - to restore the context.

**Type:** String.

**Default value:**

**Usage:**

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

---

## Sensor Control

**Note:** Some of XIMEA cameras have sensors with specific features.

### XI\_PRM\_SENSOR\_FEATURE\_SELECTOR or "sensor\_feature\_selector"

**Description:** Selects the current feature which is accessible by [XI\\_PRM\\_SENSOR\\_FEATURE\\_VALUE](#). See more at our support page, [SENSOR FEATURE SELECTOR](#).

**Type:** Enumerator.

**Default value:** XI\_SENSOR\_FEATURE\_ZEROROT\_ENABLE

**Usage:**

```
int sensor_feature_selector = 0;
xiGetParamInt(handle, XI_PRM_SENSOR_FEATURE_SELECTOR, &sensor_feature_selector);

xiSetParamInt(handle, XI_PRM_SENSOR_FEATURE_SELECTOR, XI_SENSOR_FEATURE_ZEROROT_ENABLE);
```

Value	Description
XI_SENSOR_FEATURE_ZEROROT_ENABLE	Sensor Zero ROT enable for ONSEMI PYTHON family. For camera model:MQ013xG-ON (on/off)

XI_SENSOR_FEATURE_BLACK_LEVEL_CLAMP	Black level offset clamping (value). for Camera model:MD
XI_SENSOR_FEATURE_MD_FPGA_DIGITAL_GAIN_DISABLE	Disable digital component of gain for MD family (1=disabled/0=enabled)
XI_SENSOR_FEATURE_ACQUISITION_RUNNING	Sensor acquisition is running status (0/1). Could be stopped by setting of 0. For camera model:CB,MC,MX,MT
XI_SENSOR_FEATURE_TIMING_MODE	Sensor timing mode (value depends on sensor)
XI_SENSOR_FEATURE_PARALLEL_ADC	Enables the parallel ADC readout mode, where all exposed pixels undergo dual sampling, leading to reduced readout noise at the cost of increased readout time
XI_SENSOR_FEATURE_BLACK_LEVEL_OFFSET_RAW	Sensor specific register raw black level offset (value)
XI_SENSOR_FEATURE_SHORT_INTERVAL_SHUTTER	Sensor short Interval Shutter (on/off)
XI_SENSOR_FEATURE_AUTO_LOW_POWER_MODE_AUTO	Sensor low power mode (on/off)
XI_SENSOR_FEATURE_HIGH_CONVERSION_GAIN	Enables high conversion gain feature which applies additional gain to the signal at the pixel level. This leads to a reduction in read noise and a boost in sensitivity and signal-to-noise ratio, particularly in low-light situations. Consequently, the camera exhibits superior performance in dark environments, capturing images with minimal noise and enhanced detail.
XI_SENSOR_FEATURE_DUAL_TRG_EXP_ZONE_DIVIDER_POSITION	Sensor Dual Trigger Exposure Zone Divider Position
XI_SENSOR_FEATURE_TOF_VCSEL_CTRL_VOLTAGE_MV	ToF VCSEL Control Voltage in mV

## XI\_PRM\_SENSOR\_FEATURE\_VALUE or "sensor\_feature\_value"

Description: Allows access to sensor feature value currently selected by

[XI\\_PRM\\_SENSOR\\_FEATURE\\_SELECTOR](#)

**Type:** Integer.

**Default value:** 0

**Typical range:** [ 0, 1024 ]

Is invalidated by: [XI\\_PRM\\_SENSOR\\_FEATURE\\_SELECTOR](#)

**Usage:**

```
xiSetParamInt(handle, XI_PRM_SENSOR_FEATURE_SELECTOR, XI_SENSOR_FEATURE_ZEROROT_ENABLE);
```

```
xiSetParamInt(handle, XI_PRM_SENSOR_FEATURE_VALUE, XI_ON);
```

## Extended Features

### XI\_PRM\_ACQUISITION\_STATUS\_SELECTOR or "acquisition\_status\_selector"

Description: Selects the internal acquisition signal to read using [XI\\_PRM\\_ACQUISITION\\_STATUS](#)

**Type:** Enumerator.

**Default value:** XI\_ACQUISITION\_STATUS\_ACQ\_ACTIVE

**Usage:**

```
int acquisition_status_selector = 0;
```

```
xiGetParamInt(handle, XI_PRM_ACQUISITION_STATUS_SELECTOR, &acquisition_status_selector);
```

```
xiSetParamInt(handle, XI_PRM_ACQUISITION_STATUS_SELECTOR, XI_ACQUISITION_STATUS_ACQ_ACTIVE);
```

Value	Description
XI_ACQUISITION_STATUS_ACQ_ACTIVE	Device is currently doing an acquisition of one or many frames.

## XI\_PRM\_ACQUISITION\_STATUS or "acquisition\_status"

**Description:** Returns status of acquisition.

**Type:** Enumerator.

**Default value:** XI\_OFF

**Usage:**

```
int value = 0;
```

```
xiGetParam(handle, XI_PRM_ACQUISITION_STATUS, &value, sizeof(int), xiTypeInteger);
```

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

Value	Description
XI_OFF	Turn parameter off
XI_ON	Turn parameter on

## XI\_PRM\_DP\_UNIT\_SELECTOR or "dp\_unit\_selector"

**Description:** Data Pipe Unit Selector.

**Type:** Enumerator.

**Default value:** XI\_DP\_UNIT\_SENSOR

**Usage:**

```
int dp_unit_selector = 0;
```

```
xiGetParamInt(handle, XI_PRM_DP_UNIT_SELECTOR, &dp_unit_selector);
```

```
xiSetParamInt(handle, XI_PRM_DP_UNIT_SELECTOR, XI_DP_UNIT_SENSOR);
```

Value	Description
XI_DP_UNIT_SENSOR	Selects device image sensor
XI_DP_UNIT_FPGA	Selects device image FPGA

## XI\_PRM\_DP\_PROC\_SELECTOR or "dp\_proc\_selector"

**Description:** Data Pipe Processor Selector.

**Type:** Enumerator.

**Default value:** XI\_DP\_PROC\_NONE

Is invalidated by: [XI\\_PRM\\_DP\\_UNIT\\_SELECTOR](#)

**Usage:**

```
int dp_proc_selector = 0;
xiGetParamInt(handle, XI_PRM_DP_PROC_SELECTOR, &dp_proc_selector);
xiSetParamInt(handle, XI_PRM_DP_PROC_SELECTOR, XI_DP_PROC_NONE);
```

Value	Description
XI_DP_PROC_NONE	Default empty processor
XI_DP_PROC_CHANNEL_MUXER	Channel Muxer (selected processor combines multiple input channels)
XI_DP_PROC_PIXEL_SEQUENCER	Selects pixel data output sequence
XI_DP_PROC_CHANNEL_1	Selects sensor output channel 1
XI_DP_PROC_CHANNEL_2	Selects sensor output channel 2
XI_DP_PROC_FRAME_BUFFER	Selects frame buffer memory

## XI\_PRM\_DP\_PARAM\_SELECTOR or "dp\_param\_selector"

**Description:** Data Pipe Processor parameter Selector.

**Type:** Enumerator.

**Default value:** XI\_DP\_PARAM\_NONE

**Is invalidated by:** XI\_PRM\_DP\_UNIT\_SELECTOR, XI\_PRM\_DP\_PROC\_SELECTOR

**Usage:**

```
int dp_param_selector = 0;
xiGetParamInt(handle, XI_PRM_DP_PARAM_SELECTOR, &dp_param_selector);
xiSetParamInt(handle, XI_PRM_DP_PARAM_SELECTOR, XI_DP_PARAM_NONE);
```

Value	Description
XI_DP_PARAM_NONE	Empty parameter
XI_DP_PARAM_CHMUX_CHANNEL_SELECTOR	Defines output of Channel Muxer processor
XI_DP_PARAM_CHMUX_ALPHA	Channel merger coefficient Alpha
XI_DP_PARAM_CHMUX_BETA	Channel merger coefficient Beta
XI_DP_PARAM_PIXSEQ_SELECTOR	PixSeq Selector
XI_DP_PARAM_CHANNEL_TIMING	Selected channel timing
XI_DP_PARAM_FRAMEBUF_MODE	Frame Buffer Mode
XI_DP_PARAM_FRAMEBUF_SIZE	Frame Buffer Size Bytes

## XI\_PRM\_DP\_PARAM\_VALUE or "dp\_param\_value"

**Description:** Data Pipe processor parameter value.

**Type:** Float.

**Default value:** 0.0

**Typical range:** [ 0.0, 100000.0 ]

**Is invalidated by:** XI\_PRM\_DP\_UNIT\_SELECTOR, XI\_PRM\_DP\_PROC\_SELECTOR, XI\_PRM\_DP\_PARAM\_SELECTOR

**Usage:**

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

## XI\_PRM\_GENTL\_DATASTREAM\_ENABLED or "gentl\_stream\_en"

**Description:** Control of GenTL data stream. Enabling by XI\_ON the acquisition buffering must be controlled by GenTL interface (e.g. DSAllocAndAnnounceBuffer, DSQueueBuffer)

**Type:** Integer.

**Default value:** XI\_OFF

**Usage:**

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

## XI\_PRM\_GENTL\_DATASTREAM\_CONTEXT or "gentl\_stream\_context"

**Description:** Pointer to GenTL stream context. It can be used later with GenTL buffers handling.

**Note:** See more details in the example [xiAPI-capture-50-images-gentl](#).

**Type:** String.

**Default value:** 0

**Usage:**

```
xiSetParamInt(handle, XI_PRM_GENTL_DATASTREAM_ENABLED, XI_ON);
void* stream_h = NULL;
xiGetParamString(handle, XI_PRM_GENTL_DATASTREAM_CONTEXT, &stream_h, sizeof(void*));
void* buffer_handle = NULL;
AllocAndAnnounceBuffer(stream_h, payload_size, NULL, &buffer_handle);
```

---

## User Set Control

**Note:** Parameters for for global control of the device settings. They allow loading or saving factory or user-defined settings to the camera memory.

### XI\_PRM\_USER\_SET\_SELECTOR or "user\_set\_selector"

[Description: User Set to be loaded by XI\\_PRM\\_USER\\_SET\\_LOAD.](#)

**Note:** Available only on some camera models: MX377, MJ042, MJ150.

**Type:** Enumerator.

**Default value:** 0

**Usage:**

```
int user_set_selector = 0;
xiGetParamInt(handle, XI_PRM_USER_SET_SELECTOR, &user_set_selector);
xiSetParamInt(handle, XI_PRM_USER_SET_SELECTOR, XI_US_12_STD_L);
```

Value	Description
XI_US_12_STD_L	12bit per channel STD Low Gain mode preset.
XI_US_12_STD_H	12bit per channel STD High Gain mode preset.
XI_US_14_STD_L	14bit per channel STD Low Gain mode preset.
XI_US_NONE	No preset selected.
XI_US_14_STD_H	14bit per channel STD High Gain mode preset.
XI_US_2_12_CMS_S_L	12bit per channel, 2 samples, CMS(summing) Low Gain mode preset.
XI_US_2_12_CMS_S_H	12bit per channel, 2 samples, CMS(summing) High Gain mode preset.
XI_US_2_14_CMS_S_L	14bit per channel, 2 samples, CMS(summing) Low Gain mode preset.
XI_US_2_14_CMS_S_H	14bit per channel, 2 samples, CMS(summing) High Gain mode preset.
XI_US_4_12_CMS_S_L	12bit per channel, 4 samples, CMS(summing) Low Gain mode preset.
XI_US_4_12_CMS_S_H	12bit per channel, 4 samples, CMS(summing) High Gain mode preset.
XI_US_4_14_CMS_S_L	14bit per channel, 4 samples, CMS(summing) Low Gain mode preset.
XI_US_4_14_CMS_S_H	14bit per channel, 4 samples, CMS(summing) High Gain mode preset.
XI_US_2_12_HDR_HL	12bit per channel, 2 samples, HDR High Low Gain mode preset.
XI_US_2_12_HDR_L	12bit per channel, 2 samples, HDR Low Gain mode preset.
XI_US_2_12_HDR_H	12bit per channel, 2 samples, HDR High Gain mode preset.
XI_US_4_12_CMS_HDR_HL	12bit per channel, 4 samples, CMS + HDR High Low Gain mode preset.
XI_US_2_14_HDR_L	14bit per channel, 2 samples, HDR Low Gain mode preset.
XI_US_2_14_HDR_H	14bit per channel, 2 samples, HDR High Gain mode preset.
XI_US_2_12_CMS_A_L	12bit per channel, 2 samples, CMS(averaging) Low Gain mode preset.
XI_US_2_12_CMS_A_H	12bit per channel, 2 samples, CMS(averaging) High Gain mode preset.

## XI\_PRM\_USER\_SET\_LOAD or "user\_set\_load"

Description: Loads User Set selected by XI\_PRM\_USER\_SET\_SELECTOR. User Set is list of API parameters and values, which is applied similarly as xiSetParam one by one. If setting of some parameter fails, the process of loading is aborted and error value is returned to the application. All parameters changed remains without any restore to previous state.

**Note:** Available only on some camera models: MX377, MJ042, MJ150.

**Type:** Integer.

**Default value:** 0

**Usage:**

```
int value = 0;
```

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

## XI\_PRM\_USER\_SET\_DEFAULT or "user\_set\_default"

**Description:** Selected User Set to load and make active when the device is opened. Change might affect default mode in other applications, e.g. CamTool.

**Note:** Available only on some camera models: MX377, MJ042, MJ150.

**Type:** Enumerator.

**Default value:** 0

**Usage:**

```
int user_set_default = 0;
xiGetParamInt(handle, XI_PRM_USER_SET_DEFAULT, &user_set_default);
xiSetParamInt(handle, XI_PRM_USER_SET_DEFAULT, XI_US_12_STD_L);
```

Value	Description
XI_US_12_STD_L	12bit per channel STD Low Gain mode preset.
XI_US_12_STD_H	12bit per channel STD High Gain mode preset.
XI_US_14_STD_L	14bit per channel STD Low Gain mode preset.
XI_US_NONE	No preset selected.
XI_US_14_STD_H	14bit per channel STD High Gain mode preset.
XI_US_2_12_CMS_S_L	12bit per channel, 2 samples, CMS(summing) Low Gain mode preset.
XI_US_2_12_CMS_S_H	12bit per channel, 2 samples, CMS(summing) High Gain mode preset.
XI_US_2_14_CMS_S_L	14bit per channel, 2 samples, CMS(summing) Low Gain mode preset.
XI_US_2_14_CMS_S_H	14bit per channel, 2 samples, CMS(summing) High Gain mode preset.
XI_US_4_12_CMS_S_L	12bit per channel, 4 samples, CMS(summing) Low Gain mode preset.
XI_US_4_12_CMS_S_H	12bit per channel, 4 samples, CMS(summing) High Gain mode preset.
XI_US_4_14_CMS_S_L	14bit per channel, 4 samples, CMS(summing) Low Gain mode preset.
XI_US_4_14_CMS_S_H	14bit per channel, 4 samples, CMS(summing) High Gain mode preset.
XI_US_2_12_HDR_HL	12bit per channel, 2 samples, HDR High Low Gain mode preset.
XI_US_2_12_HDR_L	12bit per channel, 2 samples, HDR Low Gain mode preset.
XI_US_2_12_HDR_H	12bit per channel, 2 samples, HDR High Gain mode preset.
XI_US_4_12_CMS_HDR_HL	12bit per channel, 4 samples, CMS + HDR High Low Gain mode preset.
XI_US_2_14_HDR_L	14bit per channel, 2 samples, HDR Low Gain mode preset.
XI_US_2_14_HDR_H	14bit per channel, 2 samples, HDR High Gain mode preset.
XI_US_2_12_CMS_A_L	12bit per channel, 2 samples, CMS(averaging) Low Gain mode preset.
XI_US_2_12_CMS_A_H	12bit per channel, 2 samples, CMS(averaging) High Gain mode preset.

---

## API parameter modifiers

**Description:** The parameter modifiers allow you to acquire more information about the camera parameters (e.g. min. or max. value). Also with certain parameters they allow direct update of these parameters without interrupting the image acquisition loop (e.g. setting of exposure and gain).

### XI\_PRM\_INFO\_SETTABLE

**Description:** Check if parameter is settable. It finishes with success when settable.

**Usage:**

```
if (XI_OK == xiSetParamInt(handle, XI_PRM_TEMP_SELECTOR XI_PRM_INFO_SETTABLE,
XI_TEMP_SENSOR_BOARD))
{
    printf("Camera supports TEMP_SENSOR_BOARD\n");
}
```

## XI\_PRM\_INFO\_MIN

**Description:** Acquire parameter minimum value

**Usage:**

```
int exp_min = 0;
xiGetParamInt(handle, XI_PRM_EXPOSURE XI_PRM_INFO_MIN, &exp_min);
float framerate = 0;
xiGetParamFloat(handle, XI_PRM_FRAMERATE XI_PRM_INFO_MIN, &framerate);
```

## XI\_PRM\_INFO\_MAX

**Description:** Acquire parameter maximum value.

**Usage:**

```
int exp_max = 0;
xiGetParamInt(handle, XI_PRM_EXPOSURE XI_PRM_INFO_MAX, &exp_max);
float framerate = 0;
xiGetParamFloat(handle, XI_PRM_FRAMERATE XI_PRM_INFO_MAX, &framerate);
```

## XI\_PRM\_INFO\_INCREMENT

**Description:** Get parameter possible increment step. The setting of value is limited to values  $\text{MinumumValue} + (N * \text{IncrementValue})$

**Usage:**

```
int height_inc = 0;
xiGetParamInt(handle, XI_PRM_HEIGHT XI_PRM_INFO_INCREMENT, &height_inc);
```

## XI\_PRM\_DIRECT\_UPDATE

**Description:** Parameter modifier for direct update without stopping the streaming. Currently `XI_PRM_EXPOSURE` and `XI_PRM_GAIN` can be used with this modifier.

**Usage:**

```
int exp_val = 0;
xiSetParamInt(handle, XI_PRM_EXPOSURE XI_PRMM_DIRECT_UPDATE, exp_val);
int gain_val = 0;
xiSetParamInt(handle, XI_PRM_GAIN XI_PRMM_DIRECT_UPDATE, gain_val);
```

# Image Buffers Queue

## Functionality

The Image Buffers is first-in first-out (FIFO) type of queue.

## Capturing



Each captured image is stored in the buffers queue. When application calls xiGetImage - the oldest image is removed from queue. Maximum number of images in queue can be set by parameter XI\_PRM\_BUFFERS\_QUEUE\_SIZE.

## Flushing the queue

The images remain in the queue until they are overwritten or flushed. The queue is flushed on one of following conditions:

- acquisition is stopped (xiStopAcquisition)
- application set some of parameters using xiSetParam:
  - Exposure (XI\_PRM\_EXPOSURE) see Note
  - Gain (XI\_PRM\_GAIN) see Note
  - Downsampling (XI\_PRM\_DOWNSAMPLING)
  - Data Format (XI\_PRM\_IMAGE\_DATA\_FORMAT)
  - Width (XI\_PRM\_WIDTH)
  - Height (XI\_PRM\_HEIGHT)
  - Offset X (XI\_PRM\_OFFSET\_X)
  - Offset Y (XI\_PRM\_OFFSET\_Y)

**Note:** Some of parameters can be changed without flushing the queue. It is possible to change parameter using modifier: XI\_PRRM\_DIRECT\_UPDATE